

UNIVERSIDADE DE LISBOA  
FACULDADE DE CIÊNCIAS  
DEPARTAMENTO DE INFORMÁTICA



## **VisuaLeague II - Animated Maps for Performance Analysis in Games**

**Tiago Alexandre Orrico Moucho**

**Mestrado em Informática**

Dissertação orientada por:  
Prof.<sup>a</sup> Dr.<sup>a</sup> Ana Paula Pereira Afonso  
e Prof.<sup>a</sup> Dr.<sup>a</sup> Maria Beatriz Duarte Pereira do Carmo

2018



## Acknowledgments

First and foremost, I would like to thank my professors, Dr. Ana Paula Afonso and Dr. Maria Beatriz Carmo, for their support throughout this project, one of the main moments of my life. I also want to thank Dr. Tiago Gonçalves for the sarcastic, but good, reviews and comments that have led to improving the quality of this project.

To my parents and sister for having supported and endured me in the most difficult moments, giving me the so needed motivation to continue. A special thank you to my beautiful niece for making me happy during the most tense moments, even though she had no idea.

To all my friends and colleagues. I appreciate the laughter, the night outs and the dinners, which made me relax and gain motivation to finish this stage. Thank you Inês Matos, Ana Espinheira, Carlos Duarte, Afonso Vicente, Ana Patrícia and Raquel Nabais for the support, company, evenings and feedback, as well as the patience they had to put up with me and to help me in this whole process.

Thank you, Inês Fernandes and Rita Capela, for always being there and never doubting me.

To all the participants in the user study. Thank you very much for your availability. All your help was extremely important to this project and, without it, it was not possible to finish.

Finally, I want to thank all the other people, friends and family who were not mentioned but who motivated me throughout this process.





*Mischief managed.*



## Resumo

O fenómeno de eSports tem vindo a aumentar ao longo dos anos e, com este, também o interesse por videojogos online, tanto por parte dos jogadores como treinadores e analistas. Um dos mais populares géneros destes jogos é o Multiplayer Online Battle Arena (MOBA) e durante uma partida existem vários tipos de eventos. Estes eventos, ou ações, podem variar desde a posição de um jogador num determinado instante de tempo ou o caminho que este percorreu de um ponto a outro, a eventos mais específicos do jogo, tais como a posição onde o jogador morreu ou o número de balas disparadas numa determinada área. Analisar este tipo de dados por jogadores, treinadores e analistas pode gerar o reconhecimento de padrões permitindo definir novas estratégias que ajudem os jogadores a melhorar o seu desempenho.

Com a evolução da tecnologia, tem-se tornado cada vez mais fácil utilizar técnicas de telemetria para registar eventos em jogos online o que, por sua vez, leva a um elevado volume de dados que pode ser recolhido ao longo do tempo. Para que se possa extrair informação relevante desses conjuntos de dados, por vezes volumosos, é fundamental explorar quais as técnicas de visualização mais adequadas.

O objetivo deste trabalho consiste em utilizar o conjunto de dados telemétricos recolhidos pela Riot Games, provenientes de partidas de League of Legends (LoL), para redesenhar um protótipo onde se explora a técnica de visualização mapas animados, melhorando o desempenho computacional deste, resolvendo problemas como a sobreposição de eventos e a limitação de só suportar a análise até dois jogadores em simultâneo. Foram aplicadas técnicas de visualização complementares para a mitigação dos problemas acima mencionados, assim como técnicas com o intuito de apresentar dados estáticos e estatísticos, complementando a análise do desempenho de jogadores feita com a principal técnica de visualização utilizada. Por fim, foi feita uma avaliação do sistema de forma a perceber se as técnicas de visualização em uso são adequadas a este tipo de análise, e de que forma a ferramenta desenvolvida compete com outras existentes no mercado, com o objetivo de fazer análises do desempenho de jogadores.

Este relatório apresenta o trabalho relacionado sobre dados espaço-temporais, assim como técnicas de visualização aplicadas aos mesmos e técnicas de visualização usadas em videojogos. É discutida a relevância dos eSports e a sua possível aplicação em áreas científicas, como visualização de estruturas moleculares, pelas técnicas de visualização

utilizadas. São apresentados diferentes géneros de videojogos, como First-Person Shooter (FPS), Real-Time Strategy (RTS) e MOBA, explorando o tipo de análise que pode ser feita entre os géneros. É feita uma breve introdução à visualização, onde se enumeram seis passos que se devem ter em conta no processo de desenho de uma visualização correta. Existindo diferentes tipos de visualização, sendo a explorada neste relatório a visualização de informação, é nesta onde recaem os tipos de dados que são analisados utilizando o protótipo desenvolvido. Para a visualização de dados espaço-temporais, são introduzidos várias técnicas, dando foco a dois métodos utilizados em projetos anteriores: o Cubo Espaço-Tempo e os Mapas Animados. Uma vez que se está a explorar técnicas de visualização adequadas para analisar jogos, são também introduzidas diferentes técnicas de visualização existentes, aplicadas diretamente dentro do jogo, como mini mapa, barras de vida e heatmaps, ou usadas por aplicações de terceiros para análise, como gráficos de barras e linhas. Ligando o trabalho já desenvolvido com o atual, são revistos dois protótipos anteriores onde já foram aplicadas as técnicas de Cubo Espaço-Tempo e Mapas Animados, referindo as vantagens e desvantagens de cada uma para a análise de desempenho de jogadores. Por fim, ainda dentro do trabalho relacionado, é efetuada uma apresentação ao jogo LoL, explicando a mecânica do jogo, cujos dados telemétricos foram utilizados para a análise e aplicação desenvolvida, explicitando que tipo de dados espaço-temporais oferece.

De seguida, é apresentado o protótipo desenvolvido juntamente com as vantagens de se construir uma aplicação web. A informação sobre a arquitetura desenvolvida é descrita em detalhe, assim como as tecnologias usadas, estando estas listadas tanto as do lado do cliente, como as do lado do servidor. Também é descrito como é que o processamento de dados foi realizado, mostrando excertos de pseudocódigo para melhor compreensão. Neste processamento de dados, é explicado como foi implementado o sistema de trajetos usado pela aplicação, assim como o tratamento efetuado sobre os dados, filtragem e reordenação, para o fácil uso e análise feita pelos utilizadores. Exibindo as funcionalidades da aplicação, é feita uma ponte entre os dados que foram processados, como estes são mostrados, e a sua possível interação com o utilizador, esclarecendo como foi realizada a sua implementação e o seu propósito. Algumas destas funcionalidades são fruto das soluções encontradas para os problemas do protótipo anterior. Outras fazem parte das técnicas de visualização utilizadas para complementar a técnica principal de visualização em uso.

Para testar a aplicação desenvolvida foram realizados dois tipos de avaliação: a primeira correspondendo a entrevistas informais e a segunda uma comparação entre a aplicação desenvolvida com outras duas existentes no mercado, o OP.GG e o Replay do LoL. Estas ferramentas são apresentadas brevemente e utilizadas na avaliação da aplicação desenvolvida, tendo sido efetuada uma comparação direta entre elas, tanto ao nível dos dados apresentados, como ao nível das funcionalidades e técnicas de visualização que usam para apresentar estes dados. Para a segunda avaliação é explicada a metodologia e

aparelhos usados na sua concretização, o tipo de participantes (jogadores casuais e profissionais da área) e as tarefas definidas que foram executadas nas três diferentes ferramentas de análise. Os resultados são apresentados, onde é discutida a precisão do sistema de trajetórias desenvolvido para a aplicação, assim como a análise feita pelos participantes para as diferentes tarefas, comparando-a entre as diferentes ferramentas utilizadas. Foi feito um levantamento das respostas finais dos participantes, onde foi perguntado o quão útil e fácil de usar é VisuaLeague II em relação às outras duas ferramentas, juntamente com a capacidade de se efetuar uma análise mais rápida. Também foi pedido aos participantes que declarassem as vantagens e desvantagens de usar VisuaLeague II em relação ao OP.GG e ao Replay do LoL.

O estudo realizado pelos participantes sugere que os utilizadores preferem visualizar dados espaço-temporais dinamicamente, ou seja, visualizando mudanças ao longo do tempo, usando mapas animados. Além disso, a visualização de dados estáticos e estatísticos, juntamente com a visualização de dados espaço-temporais, melhora a análise do desempenho do jogador. Com a análise feita pelos participantes no estudo, é possível dividir a análise em dois tipos: as análises feitas por jogadores casuais e análises feitas por profissionais (tanto jogadores como treinadores/analistas). Os jogadores casuais simplesmente descrevem o que está a acontecer na visualização. No entanto, os profissionais elaboram teorias sobre o desempenho de um jogador, dando sugestões sobre como o jogador devia ter jogado em certas ocasiões.

No geral, os participantes demonstraram uma atitude mais positiva em relação ao uso do VisuaLeague II e do Replay do LoL do que o OP.GG como uma ferramenta de análise, simplesmente porque melhores técnicas de visualização estão a ser utilizadas para visualizar dados espaço-temporais.

Trabalho futuro também é mencionado onde se explicita vários pontos em que VisuaLeague II pode ser melhorado. Uma ideia é formulada com o propósito de ser possível a análise do desempenho do jogador não apenas em um jogo, mas ser possível expandir o protótipo de forma a analisar o desempenho do jogador ao longo de vários jogos.

**Palavras-chave:** Dados Espaço-Temporais, Mapas Animados, League of Legends



## Abstract

The phenomenon of eSports has been increasing over the years as well as the interest in online video games by players, coaches and analysts. One of the most popular genres of these games is the Multiplayer Online Battle Arena (MOBA) and during a match there are several types of events. These events, or actions, can range from the path a player traveled from one point to another, to more specific events in the match, such as the position where the player died. Analyzing this type of data by these stakeholders can generate pattern recognition allowing them to define new strategies that help players improve their performance.

With the evolution of technology, it has become easier to use telemetry techniques to record events in online games, which leads to a high volume of data that can be collected over time.

The objective of this work is to use the telemetric data provided by Riot Games, from League of Legends (LoL) matches, to redesign a prototype that explores the animated maps visualization technique, improving its computational performance, solving problems such as overlapping events and analysis only up to two players. Additional visualization techniques were applied to mitigate the above-mentioned problems, as well as the addition of other techniques to present static and statistical data.

Finally, a comparison between the developed tool (VisuaLeague II) and other existing ones in the market (OP.GG and LoL's Replay) was performed to understand if the visualization techniques implemented are adequate for analyzing player's performance.

The study carried out suggests that participants demonstrated a more positive attitude towards using VisuaLeague II and LoL's Replay rather than OP.GG as an analysis tool, simply because better visualization techniques are being used to visualize spatio-temporal data. In addition, viewing static and statistical data, along with displaying spatio-temporal data, improves the player's performance analysis.

**Keywords:** Spatio-Temporal Data Visualization, Animated Maps, League of Legends





# Contents

<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xv</b>
<b>List of Listings</b>	<b>xvii</b>
<b>List of Acronyms</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Goals . . . . .	3
1.3 Contributions . . . . .	3
1.4 Document Structure . . . . .	4
<b>2 Concepts and Related Work</b>	<b>5</b>
2.1 Relevance of eSports . . . . .	6
2.2 Video Games Telemetry . . . . .	7
2.3 Visualization . . . . .	10
2.3.1 Spatio-Temporal Visualization Techniques . . . . .	12
2.3.2 Visualization Techniques Used in Video Games . . . . .	16
2.4 Previous Work . . . . .	21
2.5 League of Legends . . . . .	24
2.5.1 Game Description . . . . .	24
2.5.2 Telemetry Data . . . . .	26
2.6 Summary . . . . .	28
<b>3 VisuaLeague II</b>	<b>29</b>
3.1 Prototype Description . . . . .	29
3.2 Architecture . . . . .	31
3.3 Technologies . . . . .	32
3.4 Data Processing . . . . .	36
3.5 Prototype Features . . . . .	43

3.6	Summary . . . . .	50
<b>4</b>	<b>Evaluation of VisuaLeague II</b>	<b>53</b>
4.1	Informal Interviews . . . . .	53
4.2	Considered Applications for Comparison . . . . .	54
4.2.1	OP.GG . . . . .	54
4.2.2	League of Legends Replay System . . . . .	55
4.2.3	Comparison of VisuaLeague II with OP.GG and LoL's Replay . .	56
4.3	User Study . . . . .	57
4.3.1	Methodology and Apparatus . . . . .	58
4.3.2	Participants . . . . .	58
4.3.3	Tasks . . . . .	60
4.4	Results . . . . .	61
4.4.1	Routing System Accuracy . . . . .	61
4.4.2	Participants Analysis . . . . .	63
4.4.3	Survey Answers . . . . .	65
4.5	Discussion . . . . .	69
<b>5</b>	<b>Conclusion and Future Work</b>	<b>71</b>
5.1	Conclusion . . . . .	71
5.2	Future Work . . . . .	72
<b>A</b>	<b>Evaluation Form</b>	<b>75</b>
	<b>References</b>	<b>83</b>

# List of Figures

2.1	Indianapolis 500: The Simulation . . . . .	5
2.2	eSports cause-effect . . . . .	6
2.3	Jungle level heatmap representing player location . . . . .	9
2.4	Pathways visualization system showing data from level 7 . . . . .	9
2.5	A Space-Time Cube visualization of Napoleon’s march in Russia . . . . .	14
2.6	HealthVisB application with an interactive map and a scatterplot . . . . .	15
2.7	Analysis of player paths and movement . . . . .	16
2.8	Health and armor (suit) displayed on the bottom left of the screen . . . . .	17
2.9	World of Warcraft health bar . . . . .	17
2.10	Hit heatmap in Black Ops and GTA San Andreas Gang Turf Choropleth . . . . .	18
2.11	<i>Guild Wars 2</i> minimap . . . . .	19
2.12	League of Legends minimap over time . . . . .	19
2.13	Team match analysis and a line chart showing team gold earned . . . . .	20
2.14	Bar charts representing gold acquired and experience earned in a match . . . . .	20
2.15	Three kinds of visualization techniques in the GameVis application . . . . .	21
2.16	2D map and STC showing players’ paths and events over a match’s length . . . . .	22
2.17	Animated map at a certain time . . . . .	23
2.18	Lanes in Summoner’s Rift . . . . .	24
2.19	Summoner’s Rift detailed map . . . . .	25
3.1	Initial front page to fetch recent player matches . . . . .	30
3.2	VisuaLeague II Web Application . . . . .	30
3.3	MVC Architecture . . . . .	31
3.4	Neo4j nodes graph and nodes represented in Map . . . . .	34
3.5	VisuaLeague II Web Application . . . . .	43
3.6	Active tower and destroyed tower . . . . .	44
3.7	Skull shaped marker and its tooltip representing a champion kill . . . . .	44
3.8	Timeline’s design evolution . . . . .	45
3.9	VisuaLeague’s minimap . . . . .	46
3.10	VisuaLeague II types of notifications . . . . .	47
3.11	VisuaLeague II Filters Panel . . . . .	48
3.12	Filtered event’s timeline . . . . .	48

3.13	VisuaLeague II Information Panel . . . . .	49
3.14	Death status in information panel . . . . .	50
4.1	OP.GG Summoner Page with match view . . . . .	54
4.2	OP.GG's team gold and experience charts . . . . .	55
4.3	OP.GG's kill map and textual timeline . . . . .	55
4.4	LoL's Replay System . . . . .	56
4.5	Pie charts of whether tools were used . . . . .	59
4.6	Bar chart of LoL's Replay purpose . . . . .	59
4.7	Bar chart of OP.GG purpose . . . . .	60
4.8	Jungler's path accuracy per match . . . . .	62
4.9	Usefulness of VisuaLeague II compared with LoL's Replay and OP.GG . . . . .	65
4.10	Easiness of VisuaLeague II compared with LoL's Replay and OP.GG . . . . .	66
4.11	Faster analysis with VisuaLeague II compared with LoL's Replay and OP.GG . . . . .	66
4.12	Pros and cons of VisuaLeague II against LoL's Replay . . . . .	67
4.13	Pros and cons of VisuaLeague II against OP.GG . . . . .	68
4.14	Pie chart showing whether people would use VisuaLeague II . . . . .	68

# List of Tables

2.1	Participant Timeline . . . . .	26
2.2	Participant’s frame at a certain timestamp . . . . .	27
2.3	Types of Events . . . . .	27
4.1	Data Types Comparison . . . . .	57
4.2	Features Comparison . . . . .	57
4.3	<i>Latin square</i> model example . . . . .	58



# List of Listings

3.1	JSON file of recent matches' list . . . . .	35
3.2	Prepare route function's callback . . . . .	38
3.3	Match Data JSON content example . . . . .	39
3.4	Timeline Data JSON content example . . . . .	40
3.5	Prepare Timeline Data function . . . . .	42





# List of Acronyms

**AD** Attack Damage. 25, 63

**ADC** AD Carry. 25

**AI** Artificial Intelligence. 5, 24

**AP** Ability Power. 25, 63

**API** Application Programming Interface. 2, 10, 26, 28, 32, 34, 36, 38, 40, 62, 68, 71

**CS** Counter-Strike. 7, 17

**CSS** Cascading Style Sheets. 29, 35

**FPS** First-Person Shooter. vi, 1, 4, 7–9, 18, 19

**HCI** Human-Computer Interaction. 6

**HTTP** Hypertext Transfer Protocol. 29, 33, 38

**I/O** Input/Output. 33

**InfoVis** Information Visualization. 11, 12, 28

**JS** JavaScript. 29, 33, 35, 36

**JSON** JavaScript Object Notation. 28, 32, 34, 35, 39, 40, 51

**KDA** Kills, Deaths, Assists. 40, 45, 48, 49, 54, 55, 63, 64, 67

**LoL** League of Legends. v–vii, ix, 1–4, 6, 10, 19, 21–25, 28, 29, 37, 43, 45, 47, 53, 54, 56–59, 63–72

**MMO** Massive Multiplayer Online. 19

**MMR** Match Making Ratio. 59, 60

**MOBA** Multiplayer Online Battle Arena. v, vi, ix, 1, 4, 8, 10, 19

**MVC** Model-View-Controller. 31, 33

**NPC** Non-Playable Character. 5, 8

**RPG** Role-Playing Game. 24

**RTS** Real-Time Strategy. vi, 4, 8–10, 24, 26

**STC** Space-Time Cube. 13, 14, 21, 22, 28

**URL** Uniform Resource Locator. 39

**XP** Experience Points. 24–26

# Chapter 1

## Introduction

This chapter introduces the main motivation of this work, particularly addressing the importance of video game analytics as a research subject. The goals and contributions of this thesis are also characterized, followed by the structure of this document.

### 1.1 Motivation

Competitive gaming, or most commonly known as eSports, is a worldwide phenomenon in which players compete between them in online tournaments, through the use of electronic devices, mainly computers and consoles. It has become a real influence and fundamental element in today's young people culture (Wagner, 2006). Although eSports are only now being recognized as a sport, the term dates to the late nineties. It started with Doom<sup>1</sup> and Quake<sup>2</sup>, two First-Person Shooter (FPS) games which objective is to find the exit while annihilating monsters. In comparison, nowadays we have Paragon<sup>3</sup> and League of Legends (LoL)<sup>4</sup>, two Multiplayer Online Battle Arena (MOBA) games which objective is to destroy the enemy base to win the match. There are many other genres that take part in the eSports world which, in turn, contributed to a significant increase in the popularity of these events. With this growth, many necessities must be satisfied to improve the game or the gameplay, including the information given to players, coaches, and analysts, or to the developers' community, if that applies.

Video games are becoming more complex and its audience is getting bigger. This increase is making the interest in analyzing the player's behavior and the impact of design decisions more relevant (Wallner and Kriglstein, 2013). In order to improve the game, developers need feedback from users, often acquired in the form of questionnaires, playtesting or videotaping, to mention a few (Gagné et al., 2011; Wallner and Kriglstein, 2013). However, this collected data sometimes is incomplete or subjective and time-

---

<sup>1</sup><https://bethesda.net/en/game/doom>

<sup>2</sup><https://quake.bethesda.net/en>

<sup>3</sup><https://www.unrealengine.com/en-US/paragon>

<sup>4</sup><https://euw.leagueoflegends.com/en/>

consuming. This leads us to find other techniques that help getting complementary data, such as resorting to the player's direct gameplay (logging user actions). This process is called instrumentation giving, as outcome, telemetry data – a quantitative record of in-game player interactions (Loh et al., 2015; Moura et al., 2011). Telemetry data gathered can, however, be extensive, leading to a creation of big spatio-temporal datasets of information. Spatio-temporal datasets include all datasets consisting of location and time (Shrestha, 2014).

To analyze that data, visualization techniques can be applied. In this case, spatio-temporal information can help players, coaches and analysts: for example, to know where more deaths occur in the game, in a specific point, improving the player's performance and help defining new strategies; and developers: trying to find new ways to overcome issues associated with the gameplay, making improvements, implement new features based on the user's feedback and balance the game. However, the focus of this study falls on players, coaches and analysts. Studying this type of data can provide a complete meaning of what happened in a certain match or game as we can access information such as the player's position at a certain time, or even the path made from a point to another. We can also have specific events like when and/or where a player or an ability leveled up.

As a consequence of the increasing complexity in modern video games, the data associated with them is becoming increasingly more complex as well (Bowman et al., 2012). There is a lot of information to be processed and can be difficult to do such when using inappropriate visualization techniques. Therefore, it is extremely important to understand which visualization techniques are the most suitable when analyzing this type of spatio-temporal data, since the needs of players, coaches and analysts often diverge.

There is already previous experience in using animated maps to analyze spatio-temporal data. Vieira (2017) developed a prototype that supports the analysis of player's performance, by displaying spatio-temporal information associated to a LoL match. LoL is characterized by being played by two teams, of five elements each, aiming to destroy the enemy's base. There are specific events, such as player's deaths and towers destruction, and player's trajectories along the game environment that define a player's performance. Yet, the prototype had limitations: low performance, overlapping events, analysis only up to two players. Thus, this work intends to give continuity to the previous developed prototype for the purpose of overcoming the existing limitations, as the animated maps visualization technique gave an assumption of being a good technique for analyzing spatio-temporal data. LoL will be used again as a case study because the Riot Games (LoL developers) have a publicly available Application Programming Interface (API)<sup>5</sup> that provides information about player's trajectories and relevant match's events, being one of the few.

---

<sup>5</sup><https://developer.riotgames.com>

## 1.2 Goals

Using telemetry data collected from LoL, the primary goal of this project is to redesign the previous prototype to overcome its limitations, using animated maps to assist players, coaches and analysts in the analysis of player's performance. Being LoL popular world-wide, this leads to large quantities of data being generated and collected for analysis. Regular game updates contribute to different and interesting data results which, per se, enable different types of player's performance analysis.

To this end, the objectives are:

- Redesign of the VisuaLeague prototype in order to improve performance;
- Resolution of previous prototype limitations such as overlapping events and support for multi-champion analysis;
- System assessment with a tools comparison user study.

## 1.3 Contributions

The main contributions of this work can be summarized as follows:

- The study and discussion of previous research focused on the relevance of eSports and telemetry data gathered from them, applied to different visualization techniques in the context of video games, and how these subjects influence the type of analysis and the current state of video game analytics, in particular, the player's performance analysis.
- VisuaLeague II, a redesign of a previous prototype that uses the animated maps visualization technique in conjunction with other techniques to visualize spatio-temporal and thematic data, with the purpose of analyzing player's performance.
- A user study that compares the developed prototype with two other analysis tools, OP.GG and LoL's Replay, reviewing the more adequate visualization techniques to visualize spatio-temporal data, and different types of analysis made by players, coaches and analysts.
- An article describing the previous developed prototype and its objectives for the international conference - The 22 International Conference Information Visualisation (IV) 2018: Tiago Gonçalves, Pedro Vieira, Ana Paula Afonso, Maria Beatriz Carmo, Tiago Moucho. Analysing player performance with animated maps (Gonçalves et al., 2018).

- A submitted article describing the developed prototype during this thesis and its objectives for the international conference - International Conference on Graphics and Interaction (ICGI) 2018: Tiago Moucho, Ana Paula Afonso, Maria Beatriz Carmo. Visualization Tool for League of Legends Matches Analysis.

## 1.4 Document Structure

The following document is organized as follows:

In Chapter 2, some research that has been done regarding visualization techniques applied in the context of video games is presented. The relevance of eSports is discussed, focusing on fields of study they can have a meaning. The importance of telemetry data is presented, having three game genres as examples: FPS, RTS, and MOBA. The concept of visualization and information visualization is tackled, along with the visualization techniques used when analyzing spatio-temporal data and the visualization techniques used in video games. Some previous work already done is mentioned, giving insight on the direction this study should be going. Finally, an explanation of the game LoL is characterized while showing the telemetry data that can be retrieved from it.

Next, in Chapter 3 is present the VisuaLeague II prototype, the technologies and architectures used, along with the data processing and the implemented features, and how these fields connect and interact with each other.

Chapter 4 presents the evaluation process carried out with VisuaLeague II. The conducted informal interviews are described and the user study is presented that compares the developed prototype with two analysis tools available on the market, OP.GG and LoL's Replay, which are also explained.

The last one, Chapter 5, includes the main conclusions obtained through this work, the improvements that can be made to VisuaLeague II and new paths for future work.

## Chapter 2

### Concepts and Related Work

The application of visualization techniques is not new in the video games universe (Bowman et al., 2012). Such information takes form as health and mana bars, remaining ammunition, cooldown for certain abilities. This data is collected and presented to the players as they play the game, defining states in a time instant; it can also be shown to users through applications, such as web-based, without the need of being in-game and mostly data about a full match is represented by maps or plots. The data retrieved can be used to improve the player's gameplay and performance, by letting players, coaches and analysts analyze that data or allowing the developers to change how the game works and even using it against the player in real-time. INDY 500 is a simulation car driving game (Figure 2.1) where the Non-Playable Character (NPC) cars are not Artificial Intelligence (AI) based but actually constructed on the actual telemetry data being transmitted from the player's gameplay to the game platform, in real-time (Jutzi and Connelly, 2007).



Figure 2.1: Indianapolis 500: The Simulation<sup>1</sup>

There are other ways to analyze the players' behavior. The actual use of eSports as a mean to analyze the players as they play the game can help understand how the player

---

<sup>1</sup><https://tinyurl.com/ycl9s2y6>

plays it and define strategies or improvements to be made to the game. Usually this type of data is composed of spatio-temporal data and therefore, not only it is necessary to see the most adequate visualization techniques to visualize this type of data, but also what kind of visualization techniques are used in video games.

Thus, this chapter outlines the relevance of eSports and the different types of video games telemetry, along with an introduction to visualization and visualization techniques, both for spatio-temporal data and those used in video games. As this thesis is the continuation of two others, the work previously performed for the player's performance analysis is described, presenting the game LoL as case study.

## 2.1 Relevance of eSports

Although not being completely accepted as a sport, eSports has been changing some minds in the “traditional” sports community. Many believe that eSports cannot be considered as a sport because it doesn't measure the players' competence by their physical prowess, but that depends on the Human-Computer Interaction (HCI) applied to the game being played (Hamari and Sjöblom, 2017; Wagner, 2006). Based on this notion, Hamari and Sjöblom (2017) came up with the definition that eSports are “*a form of sports where the primary aspects of the sport are facilitated by electronic systems; the input of players and teams as well the output of the eSports system are mediated by human-computer interfaces*”. Also, those same minds have been changing given the fact that eSports are building, each year, bigger and better tournaments, with many players declaring it as their career. Another way of analyzing, in real-time, strategies of players is watching them in a match's key points which can be important to the player's performance or even spectators, coaches and analysts. Spectators can watch the show and see how players play so they can mimic their strategies and improve their performance. Coaches and analysts can analyze how players and teams perform to help improve their own team, in professional contexts. If with certain movements and strategies they can analyze players, we can also use that data to visualize patterns and more information about how it was done, why he/she did it and what effect it had on him/her and the match.

Players watch eSports for different reasons: acquiring knowledge; appreciation of player skills; social interaction, among others (Hamari and Sjöblom, 2017). This has an impact on how much telemetry data is created (Figure 2.2):



Figure 2.2: eSports cause-effect



If a video game is being promoted by eSports, more players will want to play that game. Video game streamers and/or professional teams are capable of building an audience and they, mostly, play everything their favorite streamer plays. This generates a snowball, more viewers equal more players, and the more players, the more data that is created.

The relevance of eSports lies not merely whether it can increase volumes of telemetry data gathered but also as a field of study which allow us to advance other areas of interest, sociology for instance, that are not directly related to computer gaming (Wagner, 2006).

The relation between eSports and science is even a lesser subject that is referred to. There is a great potential in applying certain knowledge from eSports to other fields of study. One example of that is how a fast-paced FPS game, like Counter-Strike (CS), influences player communication and language (Wagner, 2006). Wright et al. (2002) explain how socialization between players can shape their behavior while playing the game and how it affects their performance.

Lv et al. (2013) suggest that visualization challenges in particular areas can be overcome by looking towards how video games implement certain visualization techniques. More specifically, in the molecular sciences, only a small number of experts know how to design interactive visualization techniques. Using a video game engine, they were able to produce a stand-alone viewer capable of displaying molecular structures with powerful, artistic and illustrative rendering methods, same techniques used when building a video game.

Seeing how video games implement certain visualization techniques to aid the player in the game, it can be perceived as a future path to be followed to help other fields develop new and easier visualization techniques to read big quantities of data that don't require a large amount of programming or specific knowledge.

Section 2.3 will detail the visualization techniques that are used to transform big data into a more readable and adaptable data, focusing on the spatio-temporal type.

## 2.2 Video Games Telemetry

Telemetry data is collected with various instruments and sensors in real-time (Bowman et al., 2012). Game telemetry has become the name used when referring to all kinds of techniques that collect real-time data in video games. To analyze this data, there are applications that use a visualization approach. Many companies use telemetry to discover system crashes and bugs while also allowing to assess the virtual economy and catch cheaters (Moura et al., 2011). Collecting this data, however, requires developers to include in the source code some changes to trace and keep this information. Luckily, there are frameworks that help ease this process and when these frameworks don't apply, developers build custom solutions for game telemetry and analytics (Bowman et al., 2012).

Some of these are not available outside the game and the visualization and analysis of the data is constrained to it. Data retrieved through telemetry not only contains information about the game's end results but also events that happen during the session. This helps capturing the in-game behavior of individual players. Developers, players, coaches and analysts can benefit from this type of data (Wallner and Kriglstein, 2015):

- **Developers** — telemetry data can aid them evaluating their own game in regard to game design aspects and pedagogical effectiveness. Usability, playability, balancing, and difficulty are all aspects that can affect the player's performance and the data gathered from the player.
- **Players** — opportunity to monitor their own progress and how they perform in certain situations can help players improve the way they play the game and define new strategies, comparing them to other players as well. This can increase motivation, encourage competition and enable self-reflection.
- **Coaches and Analysts** — visualization of telemetry data can aid in monitoring and comparing the performance of their teams, helping to ensure that the players are focused or to improve the player's performance.

Of course, telemetry data may change accordingly to the genre of the game. As an example, FPS, RTS, and MOBA are introduced.

### First-Person Shooter

Using Halo 3<sup>2</sup> as a case of study, Thompson (2007) reported that Bungie<sup>3</sup>, a software developer company, built custom tools for extracting gameplay data from their game. Location of each player, when and where weapons were fired, vehicles, alien kills, and the player's death were some of the content's data. They found that certain types of NPC were far too powerful and a lava pit that many players fell into. Snapshots were taken at various points in time – five minutes, one hour, eight hours - to show the players' progress. They concluded that if people were going slow, the game might be too hard in an area but going too fast and the game was easy enough. Depending on how a player died they could check if a particular alien or gun was overpowered and could balance them. At the Jungle level, players traveled randomly through the map and with that data retrieved they built, using visualization techniques, a map showing player location at five-second intervals, each colour being a new timestamp (Figure 2.3).

---

<sup>2</sup><https://www.halowaypoint.com/en-gb/games/halo-3>

<sup>3</sup><https://www.bungie.net>



Figure 2.3: Jungle level heatmap representing player location (from Thompson, 2007)

A solution to overcome this issue was to fix the terrain to keep players from backtracking. With this example, we can see that for FPS games, getting the player and death locations can help the developers see where the game fails and how they can apply fixes to give the game a better design.

### Real-Time Strategy

To see and analyze which kind of telemetry data is gathered in RTS games, Gagné et al. (2011) studied the game Pixel Legions<sup>4</sup>. They collected information about the game level, at the beginning and at the end of the level; base movement every five seconds; and death information, including the death's location, the time it happened, the team the death belonged to and the killing team. They developed a system called Pathways to represent the data set (Figure 2.4).

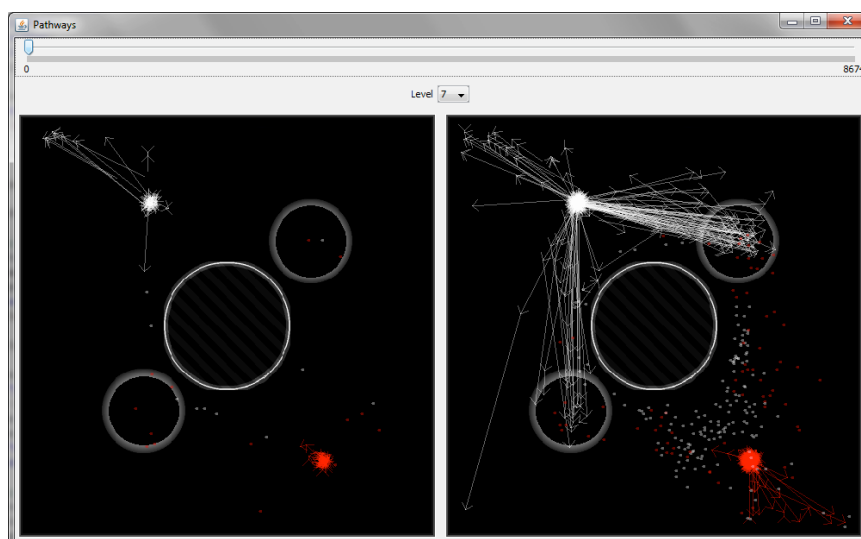


Figure 2.4: Pathways visualization system showing data from level 7 (from Gagné et al., 2011)

---

<sup>4</sup><http://pixelante.ca>

The circles represent areas that if moving team squads pass over them, it increases their power for a short period of time. On the right, for the matches that were won, it shows that players moved to those circles earlier (the visualization shows the first ten seconds of all matches). This helps answer questions proposed by Gagné et al. (2011) such as: Are the players doing what designers expected? Are there specific actions that can be associated with wins vs. losses? Are players learning how to play a level? With telemetry data, developers can see if their game works as intended and if players understand the meaning they want to demonstrate while playing the game. On the other hand, if this game were to have tournaments or a more professional context, coaches and analysts could use the telemetry data to study player's performance and improve it.

### **Multiplayer Online Battle Arena**

This genre is actually a sub-genre of RTS games (Yang et al., 2014) and as seen above, the location of players is essential when studying video game telemetry. It is important to know what the player is doing at a certain time and where, for this game genre. Four types of information can be registered when the player does or is affected, by something in a match: To whom is it happening? What is happening? Where is it happening? At what time is it happening? (Drachen and Schubert, 2013). Not only can we answer those questions, but also identify patterns in combat, player behaviors and study of complex skill learning (Moura et al., 2011; Sifa et al., 2018; Thompson et al., 2013; Yang et al., 2014). LoL, being a MOBA game, falls under this type of information and Riot Games' API offers this kind of data, although not being complete because it could compromise players' strategies<sup>5</sup>. Subsection 2.5.2 details the type of telemetry data that can be gathered using Riot Games' API.

## **2.3 Visualization**

Humans live of information. It encourages new ways to think and evolve new advancement (Khan and Khan, 2011). This information is created from different sources, are of diverse types and stored in many formats. The issue, however, is how to represent this massive data so it can be perceptible to the user, so he/she can understand and extract knowledge. The collection of information is no longer a problem but extracting valuable knowledge is.

Visualization is considered as a mental image or visual representation of an object, or scene, or person, or abstraction that is similar to visual perception (Khan and Khan, 2011). It is a graphical representation that grants the best complex ideas clearly, precisely and efficiently. The goal of visualization is to analyze, explore, discover, illustrate and communicate information in a well comprehensible form.

---

<sup>5</sup><https://tinyurl.com/yb28txnu>

Designing well-disciplined visualizations must follow a process divided into six steps (Chittaro, 2006; Khan and Khan, 2011):

1. **Mapping** — meaning how to visualize information, mapping data must transform information into graphical form under the assumption of visual features.
2. **Selection** — make a selection of data according to the given task or job. This step is directly dependent on the aim to get through visual graphics.
3. **Presentation** — how to manage, organize information in the available space on the screen effectively is the meaning of perspective presentation in visualization.
4. **Interactivity** — provide facilities to organize, explore and rearrange visualization. User-friendly interaction enables a better usage and interpretation of the data.
5. **Human Factor** — usability and accessibility are two broad categories that need to be considered when building a visualization.
6. **Evaluation** — checking if the goal is achieved or not must be the final step. After creating the visualization interface, it needs to be evaluated for its effectiveness and if it can overcome the challenges and problems in visualization (Chen, 2005; Plaisant, 2004).

Seeing that visualization consists of a process to be built upon, visualization conjointly has categories which can be divided into, according to the kind of data it represents (Khan and Khan, 2011).

There are also different types of visualizations following the periodic table of visualization<sup>6</sup>. Although there are many representations visualization can take, the type of data to be studied belongs to Information Visualization (InfoVis).

The act of gaining insight into data is InfoVis (Spence, 2001). There is not one simple definition for this category since many authors have their own. However, the most popular InfoVis representations are always mentioned when talking about this category such as Napoleon's March to Moscow or the Cholera Epidemic in London (Fekete et al., 2008; Spence, 2001). Many other examples can be found to demonstrate that in fact, InfoVis is effective and valuable (Fekete et al., 2008; Liu et al., 2014).

These helped explain how the process of InfoVis works:

- Ask a question that interests people;
- Show the right presentation;
- Let the audience understand the representation;

---

<sup>6</sup><https://tinyurl.com/w2xzw>

- Answer the question and realize how many more unexpected findings as question arise.

There are some well-known visualization techniques for data exploration such as x-y plots, line plots, and histograms (Keim, 2002). These types are adequate when visualizing small amounts and low-dimension data sets. Over the years, new techniques have been developed allowing visualization of multidimensional data. There are different data types to be visualized such as one, two or multidimensional data, hierarchies and graphs, text, among others (Keim, 2002; Shneiderman, 1996).

What lacks about these techniques is the idea of missing opportunities of interaction and distortion techniques for complementing the visualization to help the users perceive better information or focusing only on what he/she seeks. Filtering data can help the users quickly focus on their interests by eliminating unwanted items. The application of zoom when dealing with large amounts of data can be useful, allowing a variable display of the data on different resolutions, giving even more detailed data if applicable. Using multiple views (overview and detail view) at the same time can help not losing context of the primary view, giving more detail within the information space. Creating relationships between items can also be proven convenient when in need to retrieve similar data to what is being presented. Distortion techniques offer the ability to show portions of the data with a high level of detail while others are shown with a lower level of detail (Keim, 2002; Khan and Khan, 2011; Shneiderman, 1996). The exploration of large data sets is an important, but difficult problem. InfoVis techniques may help solve the problem if applied correctly, keeping in mind other existing problems, i.a. scalability, prior knowledge, and usability, mentioned by Chen (2005). As this project is dealing with spatio-temporal data, the topic that follows presents some visualization techniques that help represent this type of data.

### 2.3.1 Spatio-Temporal Visualization Techniques

Spatio-temporal data, as the name suggests, is data that depends on both time and space. Time is normally visualized using two temporal primitives: time point or time interval (Shrestha, 2014). Time points are limited to whether two events took place at the same time or one before the other. As for time interval, its usefulness lies whether events overlapped or when they started/ended, including all the questions the prior temporal primitive could answer. Visualizing space information has always been dominated by maps. Maps are a theme based graphic representation of spatial concepts (Bagrow, 2017).

Although applied in different contexts, the analysis of spatio-temporal data, such as the players' trajectories in the virtual world is not completely different from the analysis of real-world human trajectories (Drachen and Schubert, 2013), and, therefore, the results from this research area should be considered.

Several visualization techniques have been proposed to represent spatio-temporal data. Due to their simplicity, the most used techniques are based on static maps. One such approach is to use points and/or lines over the geographic area, in which each point can symbolize a visited location (Wallner and Kriglstein, 2013).

Choropleth and heatmaps are also common approaches, where maps are coloured according to certain metrics' values, e.g. death rates and popular density. While choropleth maps associate a colour to a specific location (Bowman et al., 2012), heatmaps aggregate that information and represent it with a gradient (Wallner and Kriglstein, 2015). However, these approaches require a large amount of data to be useful, and can often conceal important information due to design constraints (Wallner and Kriglstein, 2013). For example, when a 3D environment is mapped into a 2D heatmap, information related to the height is hidden.

The application of visualization techniques into the area of sports is also highly relevant as it explores many approaches for spatio-temporal visualization and is often used by professional athletes and sports institutions. Overall, the tools developed to aid in the analysis of gameplay differ greatly based on each type of sport. Despite of that, the objectives regarding their usage share several similarities, including the player's performance analysis through various metrics, e.g. points scored, number of passes; their trajectories, or the analysis and reconstruction of specific events. The application of visualization techniques enabling an in-depth data analysis was already introduced in sports, such as soccer (Perin et al., 2013; Gudmundsson and Wolle, 2014; Stein et al., 2018).

Gudmundsson and Horton (2017) provide a literature review of the most relevant works that use spatio-temporal data from team sports and identify several research questions. A relevant challenge is the lack of user studies exploring analytical questions that experts need support for and which types of techniques can be understood by them. Our main objectives go in agreement with this challenge, by studying if players, coaches and analysts are capable of reaching meaningful conclusions when analyzing player's performance using animated maps.

Consequently, Space-Time Cube (STC) and animated maps visualization techniques are introduced, as previous studies (Carreiro, 2016; Vieira, 2017) used these same techniques to analyze player's performance (Section 2.4).

### **Space-Time Cube**

A visual exploration of spatio-temporal data requires tools that can help users find answers to different types of questions. Hägerstrand (1970) suggested a three-dimensional diagram, nominated space-time cube, to show people's life histories and how they interact in space and time (Gatalsky et al., 2004). The cube has on its base a geographic representation (along the x- and y-axis), while the z-axis, corresponding to the cube's height, represents time (Andrienko et al., 2003; Gatalsky et al., 2004; Kraak, 2003; Shrestha,

2014).

Figure 2.5 shows a STC visualization of Napoleon’s march to Moscow, having a portion of the geographical map at the bottom of the cube, the dates represented along the z-axis, points representing waypoints, and a line which its width represents the number of people.

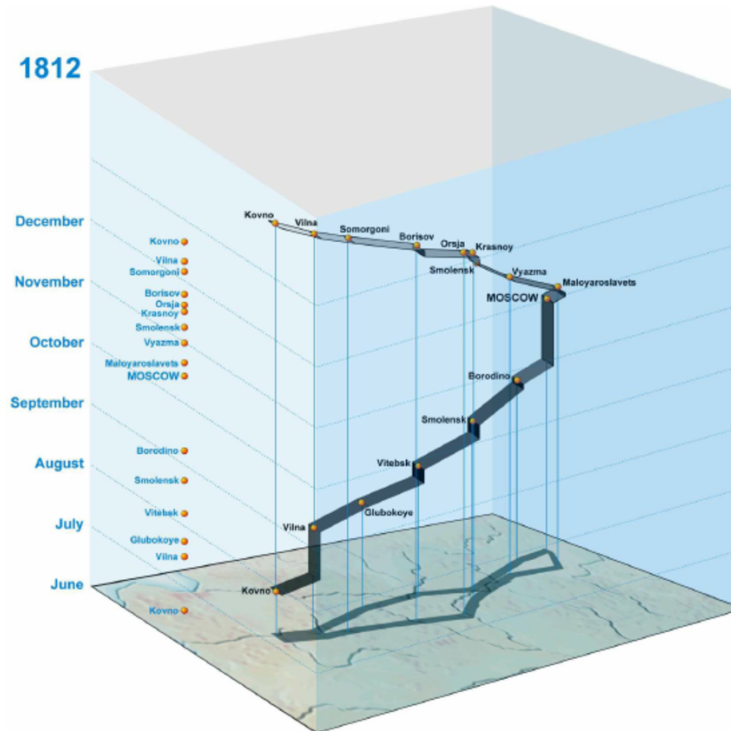


Figure 2.5: A STC visualization of Napoleon’s march in Russia<sup>7</sup>

An important feature of the STC presentation is its dynamic linking with a map as well as graphical displays of other types. This kind of visualization could be connected with isosurfaces since it inherently supports three dimensions (Shrestha, 2014). Andrienko et al. (2003) discusses the existing techniques and the queries this visualization supports, such as filtering and labeling. The main advantage of using a 3D representation space is the possibility of integrating additional visual information into the representation. However, the challenge when visualizing data with temporal dependencies is coping with large data sets, which may result in overlapping and cluttered information (Tominski et al., 2005).

### Animated Maps

Animated maps, also called movie maps or change maps, are used to depict geographic change and processes over time, contrarily to static maps that present all of their information simultaneously (Harrower and Fabrikant, 2008). They display a sequence frames,

<sup>7</sup><https://tinyurl.com/y9xwuhvg>



usually in a single view, taking advantage of the computer’s capability to rapidly update its contents. One study showed that animated representation allows users to more often correctly identify if particular types of patterns were present than using static small-multiple representation (Griffin et al., 2006), this being a presentation of multiple static maps simultaneously.

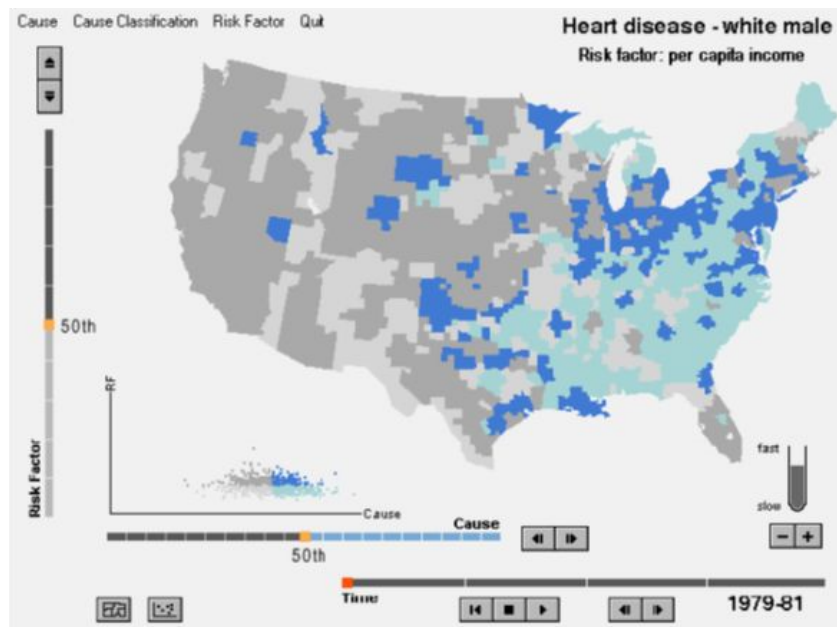


Figure 2.6: HealthVisB application with an interactive map and a scatterplot<sup>8</sup>

The Figure 2.6 presents the HealthVisB application with a full set of available controls – spatial, temporal and attribute features. On the map, blue is used to indicate higher mortality rates and dark shades correspond to higher values for the risk factor. The designing process should consider existing problems, and their solutions, for creating an effective animated map (Harrower, 2003). One point that must be mentioned is the use of additional information when using animated maps. Sometimes, animated maps are not enough to gain insights into highly complex data, so the usage of multi-component animated maps (additional diagrams and graphs) is a useful tactic (Opach et al., 2014). Figure 2.6 uses this multi-component approach, adding a scatterplot — risk factor by cause — to an animated map. Although animated maps sound like a good visualization technique to use when handling spatio-temporal data, there is still work to do in order to realize the full potential of animated maps; “*we need to understand for what kinds of representational tasks are well suited and how variations in the design of animated maps impact our ability to communicate and learn*” (Harrower and Fabrikant, 2008).

<sup>8</sup>Adapted from [http://images.slideplayer.com/24/7541482/slides/slide\\_22.jpg](http://images.slideplayer.com/24/7541482/slides/slide_22.jpg)

### 2.3.2 Visualization Techniques Used in Video Games

Video games are becoming more complex and are reaching a bigger audience. Consequently, there is an increasing interest in methods to analyze player behavior and the impact on game decisions (Wallner and Kriglstein, 2013). Such data would inform game and level designers of game design issues that need to be fixed or improved upon (Moura et al., 2011). While it is a good idea to analyze this data, correct visualization techniques should be picked to get the most valuable information. Moura et al. (2011) provided some insight about visualizations techniques proposed to represent players' path (Figure 2.7) or specific events, such as death and spawn points, using heatmaps. They created an application to see many types of information, by using circles to represent time spent in each area of a map, or interaction with characters, in the *Dragon Age Origins*<sup>9</sup> game. All the map visualization went along with bar plots, complementing the information shown. Figure 2.7 represents a map of the game showing the circles mentioned before, while also showing the lines representing the paths taken by the players. Below the map is the location of a bar plot, showing connections between each location.



Figure 2.7: Analysis of player paths and movement (from Moura et al., 2011)

The last representation referred to visualization techniques applied outside the game, when they are the most common. However, some of them are actually used in-game. The most common one is representing visually information about health, stamina, and ammo states. Health is a critical information for the player, it determines how much room is left for mistakes (Zammitto, 2008). In the game *Half-Life 2*<sup>10</sup>, a full health status is represented by 100 and the yellow colour (Figure 2.8). When the user hits a lower health state, the colour turns red. This colour-coded technique corresponds to established conventions, meaning red for danger.

<sup>9</sup><http://dragonage.bioware.com/dao>

<sup>10</sup><http://orange.half-life2.com>



Figure 2.8: Health and armor (suit) displayed on the bottom left of the screen (from Zammito, 2008)

One other example of representing health stats is by using bars. In *World of Warcraft*<sup>11</sup>, health bars do not use the colour-coding technique as many games do. The colour of the bar is always the same, green for health and yellow for energy, but a length bar cue is applied (Figure 2.9). This is a good technique as colour-blinded people may not perceive certain shades of green.

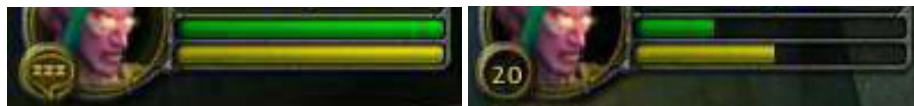


Figure 2.9: World of Warcraft health bar<sup>13</sup>

Being two-dimensional graphical representations of data, heatmaps show values of a variable as colours (Bojko, 2009). They can represent various types of data, such as usage (clicks, key presses), accuracy or visual attention. The intuitive nature of the colour as a scale relating to temperature minimizes the amount of learning necessary to understand it. As yellow is warmer than green, orange being warmer than yellow, and red representing hot, it is not difficult to figure out that the amount of heat is proportional to the level represented by the analyzed variable. Besides showing gaze interaction<sup>12</sup>, heatmaps can present information about the most occupied places in a stadium by the players or frequent walkways (Goldsberry, 2012; Pileggi et al., 2012), or show firing locations for each type of weapon in the game CS: Global Offensive<sup>13</sup>.

<sup>11</sup><https://worldofwarcraft.com/en-us/>

<sup>12</sup><http://www.ionadas.com/research-google-places-heatmaps/>

<sup>13</sup><http://blog.counter-strike.net/science/maps.html>

Bojko (2009) refers that even though heatmaps are very compelling and easy to understand, they should be used with caution and follow the correct guidelines.

Heatmaps can also be applied in-game to convey more information to the player, FPS genre being the most common. Figure 2.10 (left) shows a hit location heatmap on a person's body representing the most common spots shot with the red colour.

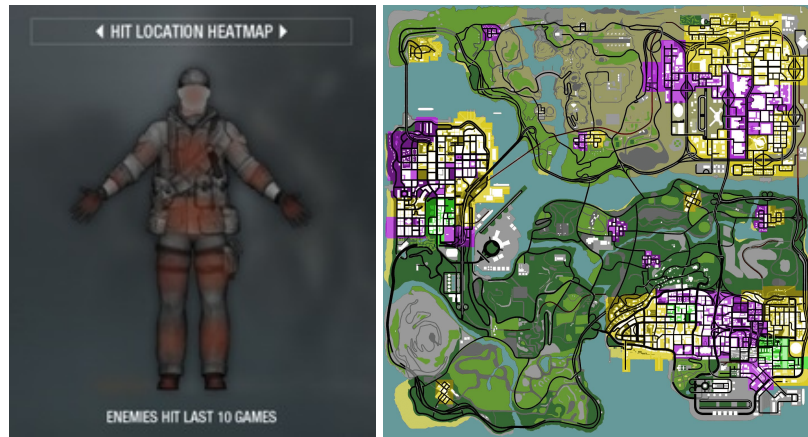


Figure 2.10: Hit heatmap in Black Ops<sup>14</sup>(left), GTA: San Andreas Gang Turf Choropleth<sup>15</sup>(right)

Figure 2.10 (right), shows a choropleth map representing gang turfs in the game *GTA: San Andreas*<sup>16</sup>. Purple represented the Ballas gang, green the Grove Street Families gang, and Yellow the Los Santos Vagos gang. The objective was to take over the enemy gang territories to give the player more respect and money<sup>17</sup>.

Minimaps are a common solution in video games to offer sense of direction in space and context over the world the user is playing. Normally the player is restricted by a view close to its character and has to resort to a bigger map visualization in order to check where to go next.

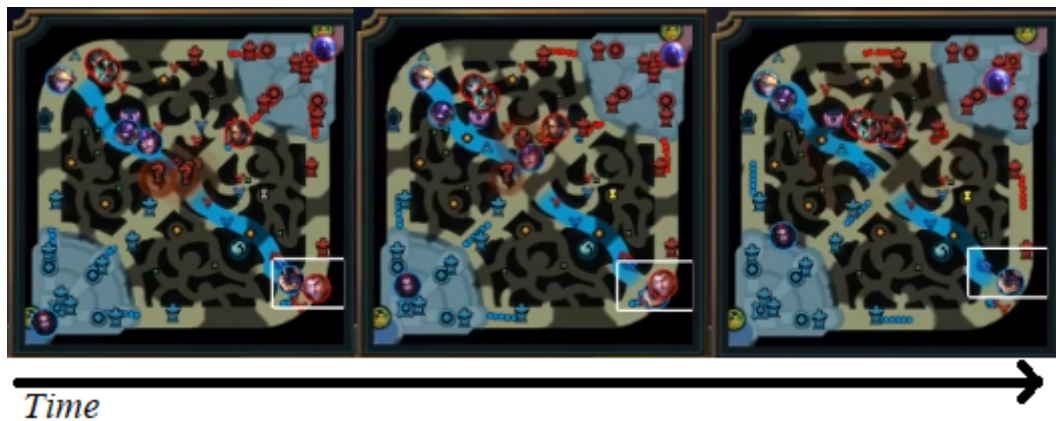
A minimap proposes a small solution to avoid the process of changing the view to a bigger map and stop the player from playing. Different minimaps representations are showed in Figure 2.11 and Figure 2.12, one representing context of a map's portion and the other representing context of the full map, respectively.

<sup>15</sup> Adapted from <https://tinyurl.com/y7blzgdh>

<sup>16</sup> <https://i.ytimg.com/vi/4DkWs81nNWE/maxresdefault.jpg>

<sup>16</sup> <http://www.rockstargames.com/sanandreas/>

<sup>17</sup> [http://gta.wikia.com/wiki/Gang\\_Warfare\\_in\\_GTA\\_San\\_Andreas](http://gta.wikia.com/wiki/Gang_Warfare_in_GTA_San_Andreas)

Figure 2.11: *Guild Wars 2*<sup>18</sup> minimap<sup>19</sup>Figure 2.12: LoL minimap over time<sup>20</sup>

Certain visualization techniques differ between game genres. When in one case, a colour-coding technique might be enough in the FPS genre to state the player's health, in the Massive Multiplayer Online (MMO) or MOBA games, a length bar is enough (Zammitto, 2008).

Whether or not visualization techniques are employed inside the game, there is always a necessity of making applications that are game-independent. Some examples are *OP.GG*<sup>21</sup> and *LolKing*<sup>22</sup>, web applications for visualization information about the player, their statistics in LoL, and champion build guides.

<sup>19</sup><https://www.guildwars2.com/en/>

<sup>20</sup>Adapted from <https://tinyurl.com/yaaph9ww>

<sup>20</sup>Adapted from <https://www.youtube.com/watch?v=jdmtRnH5VGE>

<sup>21</sup><http://euw.op.gg>

<sup>22</sup><http://www.lolking.net>



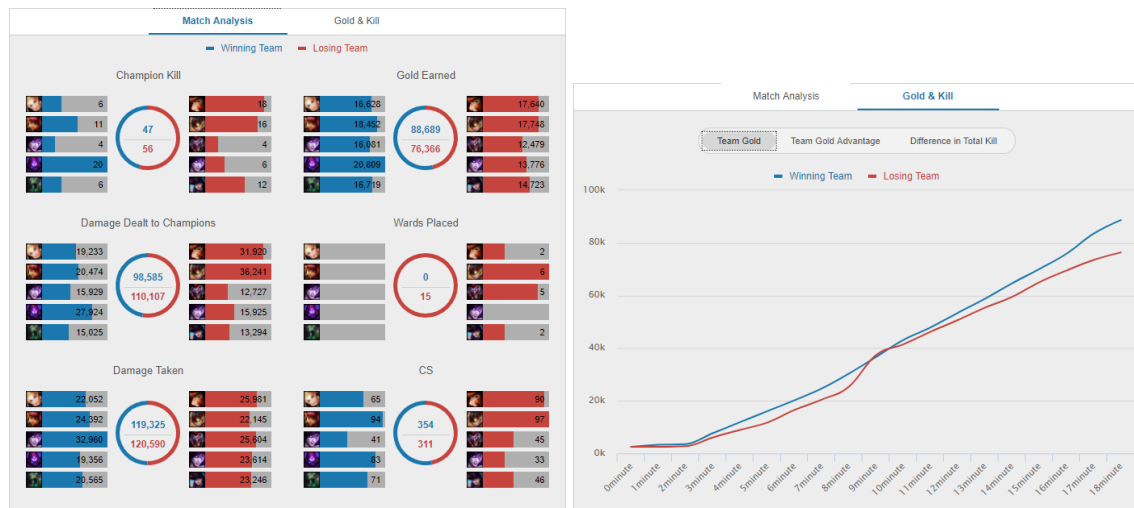


Figure 2.13: Left, a team match analysis based on various factors. Right, a line chart showing team gold earned<sup>23</sup>

Figure 2.13 show three different visualization techniques being used to represent statistic data. The first one is a variation of the bar plot; the second one, is a variation of the doughnut pie chart; and lastly, we have a typical line chart on the right. Figure 2.14 presents two bar charts indicating the gold acquired and experience earned in a match.

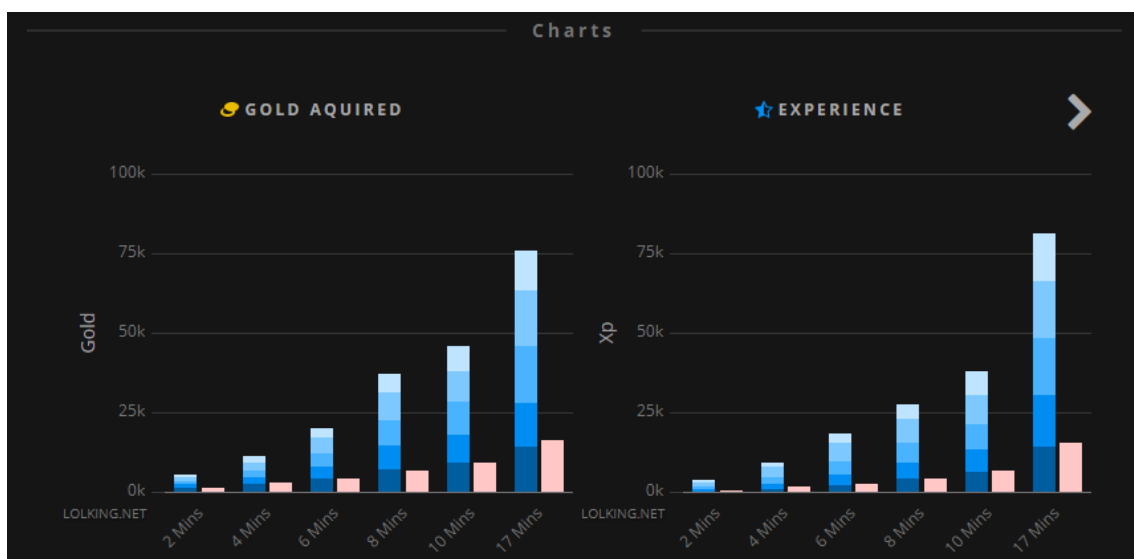


Figure 2.14: Bar charts representing gold acquired and experience earned in a match<sup>24</sup>

Most of the information presented is concerned to statistical data, where the player can see how he/she performed alongside his/her teammates or against his/her enemies.

<sup>23</sup> Adapted from <http://euw.op.gg>

<sup>24</sup> Adapted from <http://www.lolking.net>

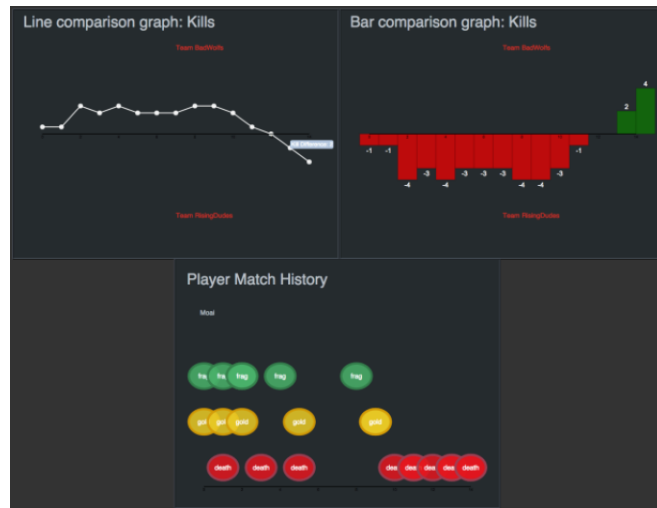


Figure 2.15: Three kinds of visualization techniques in the GameVis application (from Feitosa et al., 2015)

*GameVis* is also a web application for visualizing gameplay data, only in this case using *Dota 2*<sup>25</sup> as source, where the usage of different types of line charts, bar plots, and end match results offer good view points in visualizing game metadata (Feitosa et al., 2015).

The visualization of this type of data does not simply use one visualization technique to represent them. To get a rich and effective view of the information, some visualization techniques must be paired with each other to provide better insight over the data. However, the majority is focused on the display of aggregated statistics. This makes the detection of valuable spatio-temporal details difficult, if not impossible (Li et al., 2017) and thus, it is often necessary to combine them with map visualization techniques, providing a more informative context (Drachen and Schubert, 2013).

## 2.4 Previous Work

This thesis follows the development of visualization techniques to study spatio-temporal data in order to understand if it can help in the analysis of player's performance in games, using the game LoL as case study. Both STC and animated maps were studied in previous projects and are introduced in this section.

### Space-Time Cube

Carreiro (2016) developed a prototype where the users could choose to visualize a match of LoL in 2D, representing a map, or 3D, representing a STC. These representations depicted players' trajectories and the most relevant events in a LoL's match. In Figure 2.16

<sup>25</sup><http://www.dota2.com/play/>

we are able to visualize both representations, the STC, and the 2D map, while analyzing spatio-temporal data from a LoL match. Blue and red colours represent both teams, and their icons — skulls for player deaths and towers for tower destruction — match the events occurred in the match. The lines represent the path made by the player.

A user study was conducted to analyze a player's performance using the developed prototype against the *match-history*, an official LoL web page analysis tool. The study showed that average gamers, who do not have experience with data analysis, were interested in viewing spatio-temporal data associated with their gameplay. This led to a personal motivation of players, inciting them to take time in analyzing their own behavior. The results of usability tests show that these techniques can be used by the average gamer as an adequate way of visualizing paths and player's events.

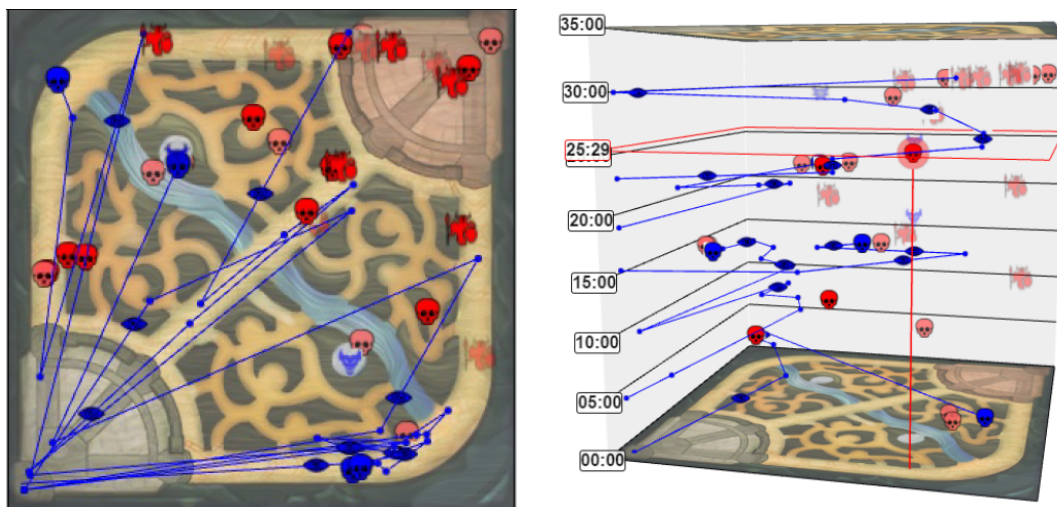


Figure 2.16: 2D map (left) and STC (right) showing players' paths and events over a match's length (adapted from Carreiro, 2016)

The STC proved to be an interesting technique, although many users reported that they would prefer sticking to the 2D view. As players are used to relying on the minimap (similar to the 2D map) while in-game, it is normal they would favor a familiar visualization. Carreiro (2016) stated that a different visualization technique should be considered, like animated maps, since the preferred view was the 2D map. Also, they would help detect movement patterns and events inside the game. A more complex representation should be considered, giving more information simultaneously about more players and game events. Finally, more studies should be made in order to enrich the analysis obtained previously and to assert their veracity.

### Animated Maps

In another project (Vieira, 2017), the animated maps visualization technique was explored to analyze player's performance. Vieira (2017) developed a web application to visualize



players' trajectories, game events and thematic information in a LoL's match to help users in the player's performance analysis. Figure 2.17 presents information about two champions, at 17 minutes and 16 seconds in the match, while also showing a death event where the champion from the blue team killed an enemy. Some tower icons are missing, indicating that they were already destroyed at that time.

The results suggested that animated maps are highly adequate to convey spatio-temporal data to users in the context of player performance analysis in video games. Furthermore, incorporating the spatio-temporal components of data using visualization techniques is essential when evaluating player performance. Users demonstrated a positive attitude towards the animated maps and wished to be able to continue using this approach to interact with spatio-temporal data during their analysis.



Figure 2.17: Animated map at a certain time<sup>26</sup>

Even though animated maps have been proven as a good technique to visualize spatio-temporal data in games, there were some issues. Overlapping events would occur and made it difficult to discern certain situations while analyzing the interaction. This application would only let visualize two champions simultaneously since it helped with the overlapping situation. The addition of other types of visualizations was recommended as players would like to analyze static and statistical information while interacting with the animated map. The application offered already some types of static information visualization, however, users felt like there should be more options. The future work mentions that for a good experience, the problems above should be fixed in a possible next iteration, adding to the study that, in fact, animated maps are really useful when analyzing player's performance in games (Vieira, 2017).

<sup>26</sup>Adapted from Vieira (2017)

## 2.5 League of Legends

LoL is a fast-paced, competitive online game that blends the speed and intensity of a RTS with Role-Playing Game (RPG) elements. Two teams, of five players each, compete with each other to defeat the opponent's nexus. With many champions to choose from, frequent updates and tournaments, LoL offers endless replayability for players of every skill level.

### 2.5.1 Game Description

There are three lanes, top, middle, and bottom (Figure 2.18), that connect the player's base to the enemy's. These roads serve as the means of engaging the enemy team. To win a game the player needs to push down their lane into the enemy base and destroy the main structure (nexus) at the center of their base.



Figure 2.18: Lanes in Summoner's Rift<sup>27</sup>

Minions are AI-controlled soldiers that spawn at the nexus and march down each lane towards the enemy base, attacking enemies they encounter along the way. When minions die, they give champions Experience Points (XP). However, only giving the last hit on a minion rewards the champion with gold.

Turrets, or towers, are defensive structures that protect each lane at even intervals hitting enemies that come within range. Some other structures, known as inhibitors, are located where each lane meets the base on both sides of the map. Destroying an inhibitor

<sup>27</sup><http://forums.na.leagueoflegends.com/board/attachment.php?attachmentid=884466>

creates more powerful minions and give players access to destroy the nexus and win the game. These inhibitors respawn after five minutes of being destroyed.

At the center of the each base lies the nexus. Surrounded by two turrets, the nexus is the structure each team needs to destroy from their opponent. Once a team has destroyed the enemy nexus, the game is over and that team wins (Figure 2.19).

LoL matches have no time limit, but they generally last between 30 to 45 minutes and can be divided into three-stages: early-game (until 15 minutes), mid-game (until 25 minutes) and late-game (until the end). However, these are not certain values as matches differ from each other.



Figure 2.19: Summoner's Rift detailed map<sup>28</sup>

Between the lanes lies the jungle, an area filled with neutral monsters. These monsters won't attack the enemy base, but they can be defeated for additional gold and special temporary buffs. A player may take on the role of Jungler — player role that plays in the jungle — and use these areas as his primary source of income. Junglers typically roam between the lanes, coming to the aid of allies in the lane when needed. There are also special monsters in the jungle, such as Baron Nashor and Dragons, that rewards teams with gold and bigger buffs.

Champions have specific classes, like Mage or Fighter, indicating the tendency of the champion's abilities: magical, Ability Power (AP); or physical, Attack Damage (AD). Each player uses an unique champion to fulfill one of the five roles: Top, Jungler, Mid, AD Carry (ADC) and Support. The player's champion earns gold and XP by taking down enemy units and structures. Killing a minion or champion, assisting in a champion kill,

<sup>28</sup>Adapted from <https://guides.gamepressure.com/lol/gfx/word/1057793420.jpg>

and acquiring certain runes or items are some of the ways the player can earn gold and XP. The player can use his/her gold to purchase items from the shop located near the base. Items grant bonuses when they are purchased, improving the player's champion abilities and stats.

The players can't see the full map. They have, however, a minimap where they can follow their teammates but can't see the enemy's champions because of the fog of war. The fog of war is a common hallmark of the RTS genre which is represented visually as a dark shroud over the terrain. To give sight, players must put wards on the map or use certain abilities or items.

### 2.5.2 Telemetry Data

The information given by the Riot Games' API consists of spatio-temporal and thematic data. They offer other types of data, but in this case, it is irrelevant, not complementing the primary data. Most of the spatio-temporal data is referred in the match endpoint. Data retrieved from a match can state, for example, the damage dealt to champions in a game (this damage can also be separated into true, physical and magical damage). An average of gold gained, minions killed, and damage taken all per minute can also be accessed in the participant timeline (Table 2.1).

Table 2.1: Participant Timeline

<i>Name</i>	<i>Description</i>	<i>Example</i>
<i>csDiffPerMinDeltas</i>	Minions score difference per minute	-2.44
<i>goldPerMinDeltas</i>	Gold gained per minute	295
<i>xpDiffPerMinDeltas</i>	Experience difference per minute	-40.15
<i>creepsPerMinDeltas</i>	Minions killed per minute	5.9
<i>xpPerMinDeltas</i>	Experience gained per minute	335.3
<i>damageTakenDiffPerMinDeltas</i>	Damage taken difference per minute	72.45
<i>damageTakenPerMinDeltas</i>	Damage taken per minute	288.2

The fields where the difference is calculated is based on the player versus the enemy he/she is confronting. For example, *xpDiffPerMinDeltas* is obtained by subtracting the *xpPerMinDeltas* of the player with the enemy's value, to give an insight of the advantage, or disadvantage, they have on one another.

Most importantly, when accessing the timeline in a match (not to be confused with participant timeline), for every 60 seconds (60000 milliseconds intervals) it is given the position of the player (spatial) per minute (temporal). Additionally, we can get details on how much gold a champion has, his/her level, minions killed, and experience gained with every interval (Table 2.2).

Table 2.2: Participant's frame at a certain timestamp

<i>Name</i>	<i>Description</i>	<i>Example</i>
<i>totalGold</i>	Participant's total gold	24809
<i>teamScore</i>	Team score of the participant	0
<i>participantId</i>	Participant ID	5
<i>level</i>	Participant's current level	18
<i>currentGold</i>	Participant's current gold	6041
<i>minionsKilled</i>	Number of minions killed by the participant	451
<i>position</i>	Participant's position	x: 5883, y: 1085
<i>xp</i>	Total of experience gained by the participant	30323
<i>jungleMinionsKilled</i>	Number of jungle minions killed by the participant	39

Apart from this, there are certain events which are recorded in the data set, presenting specific timestamps of when and where those events occurred. Only three events contain the champion's position given the timestamp. One example for omitting this value is when the champion buys an item, there is a restriction that he/she must be in his/her base to do it, so the position is not logged (Table 2.3).

Table 2.3: Types of Events

<i>Event Type</i>	<i>Description</i>	<i>Extra fields</i>
<i>CHAMPION_KILL</i>	A champion was killed	killerId, victimId, assistingParticipants, position, type
<i>WARD_PLACED</i>	A ward was placed	type, creatorId, wardType
<i>WARD_KILL</i>	A ward was destroyed	Type, killerId, wardType
<i>BUILDING_KILL</i>	A building was destroyed	killerId, buildingType, towerType, teamId, assistingParticipants, position, type, laneType
<i>ELITE_MONSTER_KILL</i>	An elite monster was killed	killerId, position, monsterType, monsterSubType, type
<i>ITEM_PURCHASED</i>	An item was bought	itemId, type, participantId
<i>ITEM_SOLD</i>	An item was sold	itemId, type, participantId
<i>ITEM_DESTROYED</i>	An item was destroyed	itemId, type, participantId
<i>ITEM_UNDO</i>	An item purchase was undone	afterId, type, participantId, beforeId
<i>SKILL_LEVEL_UP</i>	A skill was leveled up	skillSlot, levelUpType, type, participantId

Besides of the spatio-temporal information shown above, Riot Games' API provides static data that contains information about champions and descriptions, among others. All the data can be accessed via a web service and it is provided in the JavaScript Object Notation (JSON) format. There is also an access restriction implemented by the API in order to minimize abuse, to maintain a high level of stability, and to protect the underlying systems that back the API from being overloaded, e.g., 500 requests per 10 seconds when accessing the match endpoint<sup>29</sup>.

## 2.6 Summary

The analysis of the related work reveals how important the relevance of eSports is to analyze players to improve their performance by not only watching tournaments, but also how the popularity increases the number of players in a game, which in turn increases the telemetry data generated used in the analysis.

With this popularity in games, scientific areas seek to perceive certain players' behaviors and social skills, or even using techniques in game development to better visualize specific data, such as molecular structures.

The different uses of telemetry data by stakeholders and how they benefit from it was presented, introducing three different game genres. It was possible to see that, regardless of the genre, players' trajectories and specific events, such as player's deaths, are often studied.

The meaning of visualization was explored, giving focus to InfoVis, defining a 6-step process for a well-designed visualization. Different visualization techniques can be applied to improve the visualization of data, especially the visualization of spatio-temporal data. Thus, the most common ones have been mentioned, introducing, particularly, two visualization techniques, the STC and animated maps, that have been explored in past projects. Also, different visualization techniques used in games or by third-party analysis applications have been studied.

This thesis is the follow-up of another two past projects. These projects, that are introduced, used visualization techniques already explained previously, namely the STC and animated maps. The previous work using animated maps suggested that this visualization technique is really useful when analyzing player's performance, but it had limitations. In order to solve those limitations, the main objective of this thesis is to redesign the previous prototype, using the same technique.

Finally, the LoL game was introduced, along with the telemetry data it offers, being used as case study of the developed application, presented in Chapter 3.

---

<sup>29</sup><https://developer.riotgames.com/rate-limiting.html>



# Chapter 3

## VisuaLeague II

Placing the focus on the developed application, Section 3.1 provides a brief description of the developed prototype, with advantages of building a web application. Section 3.2 illustrates the architecture and Section 3.3 the technologies used, providing insight on how helpful they are, overcoming the limitations from the previous prototype. Data treatment and analysis is discussed in Section 3.4, establishing a connection with the application's features that use the processed data in Section 3.5. Design decisions are also mentioned.

### 3.1 Prototype Description

VisuaLeague II is a web application where a user can visualize a match that has taken place, independently of whom participated in it. The main goal is to assist players, coaches and analysts in the player's performance data analysis in LoL. Since LoL is very popular worldwide, the generated data quantity scales with it, making LoL a suitable candidate for study and research.

With the previous prototype limitations (Section 2.4), there was a need to build a new application, where it could offer more information being displayed at the same time, while also being reliable and fast, to find out whether this is a good analysis tool and how it compares with others in the market.

Building a web application has many advantage points:

- It can be accessed by any browser regardless of the platform.
- There is no need to be downloaded and installed in the user's system.
- Can be developed by common programming languages, like Hypertext Transfer Protocol (HTTP), Cascading Style Sheets (CSS) and JavaScript (JS).

When accessing VisuaLeague II, the initial screen shows an input area where the user can search for a player's match list in a world region. It is then presented information about the player, each match (result, scores and date), and a button to access the match's visualization, as seen in Figure 3.1.



Figure 3.1: Initial front page to fetch recent player matches

In the visualization view (Figure 3.2), a new tab is opened where the application will be displayed after being loaded. VisuaLeague II focus on the animated maps visualization technique, which covers 80% of the screen, showing players' positions, pathing progress and events (i.a. players' deaths, towers destruction). The remaining is reserved to display static and statistical data, enhancing the visualization and updating as game time goes by. Some of the features are hidden, such as the filters panel and minimap, as Figure 3.2 represents how users are greeted when opening VisuaLeague II, and these features are only shown through user interaction.



Figure 3.2: VisuaLeague II Web Application



Figure 3.5 in Section 3.5 shows VisuaLeague II with all these features being represented. All of the design and user experience decisions were based on iterative and informal interviews with players. The architecture, technologies, data processing and features are discussed in the following sections.

## 3.2 Architecture

A Model-View-Controller (MVC) architecture was used for being easy to deal with, creating simple layers between each programming section. Figure 3.3 illustrates this architecture and how it was applied.

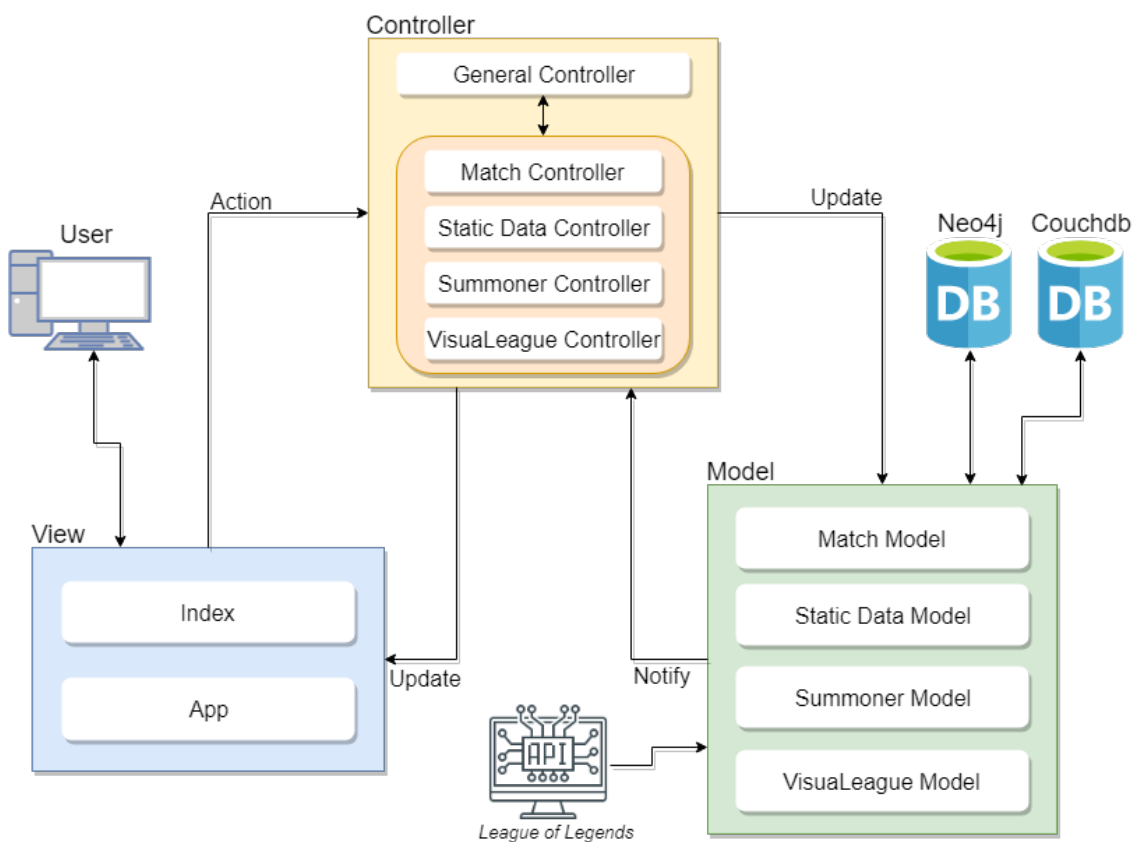


Figure 3.3: MVC Architecture

### View Section

- The Index view is responsible for the initial front page where players can access matches (Figure 3.1).
- The App view is responsible for the visualization view (Figure 3.2).

### Controller Section

- A General Controller which commands all the other defined controllers and responds to the view.
- The remaining controllers communicate with their respective models, passing that information to the General Controller.

### Model Section

- The Match Model — all data regarding the match is handled here. This model gathers content from the API. The content can be about a single match, if used in the visualization view, or a list of recent matches, to be used in the front page. Some of the content is persisted in Couchdb, like the recent matches, for later reference.
- The Static Data Model — champions, items, maps, constants and runes' lists are all part of the static data. These are referred as static because they are rarely updated. This is where the API has more time restrictions, forcing requests to be scarce.
- The Summoner Model — fetching a summoner is done by this model. The search is only made with the summoner's name but it could also be used the summoner's id. The request's result is persisted in Couchdb for later reference as well.
- The VisuaLeague Model — being the main application model, this is where all the pre-processing regarding the visualization is realized. The routing system, durations, stats over time, events are all processed here. After everything is dealt with, the content is saved in JSON, ready to be used by the front-end. Neo4j only communicates with this model as it is where the routing is formed, later discussed.

There are two databases in use, Neo4j and Couchdb. This architecture is implemented using Express, a web framework for Node.js, which in turn is the server's run-time environment. These technologies are explained in the following section.

## 3.3 Technologies

To build VisuaLeague II, some technologies were analyzed to determine which ones would help overcome the previous known limitations (Section 2.4). The following technologies solved those limitations.

### Node.js

One of the previous prototype's limitations was low performance. This defined a total redesign of VisuaLeague II. The prototype was planned to be a web application, possibly

requiring many JS libraries. To minimize the transition between front-end and back-end, while also improving performance, Node.js was elected as the server's runtime. Node.js<sup>1</sup> is a JS runtime built on Chrome's V8 JS engine. It has an event-driven architecture capable of asynchronous Input/Output (I/O) which aims to optimize throughput and scalability in web applications.

## Express

In Section 3.2, a MVC architecture was presented. To built this architecture, a web framework for Node.js was used. Express<sup>2</sup> is a lightweight web framework that helps organize a web application into a MVC architecture on the back-end. It manages routes, requests and views.

## Neo4j

Having Figure 3.2 as reference, there are some walls that define available paths for players to roam. This walls are impossible to go through, except when using special abilities. To implement the players' pathing, since we only have position data every minute or in occasional events, we decided to use a node approach. Initially we researched possible ways to make a better, authentic, routing system. However, specific and browser-independent tools were required, not fitting the design system planned, that being a web application. Overviewing these aspects, we opted to use a graph database that can help us achieve our pathing system.

Neo4j is a graph database implemented in Java that uses Cypher Query Language through a transactional HTTP endpoint<sup>3</sup>. Nodes can have attributes, as can relationships between nodes, enabling complex queries when requesting for paths or even nodes. Neo4j has already a built-in function to determine the shortest path, although it supports plugins that include all sort of functions.

Figure 3.4 shows two types of views, the left one corresponds to a browser view of the current database state, showing the created nodes and how they are connected with each other. The right one shows the nodes placed on the used map, pinpointing the exact location and also the connection between nodes. We can now see the possible paths to roam, avoiding the walls completely. There are 129 nodes and 211 relationships in total.

---

<sup>1</sup><https://nodejs.org/en/>

<sup>2</sup><https://expressjs.com>

<sup>3</sup><https://neo4j.com>

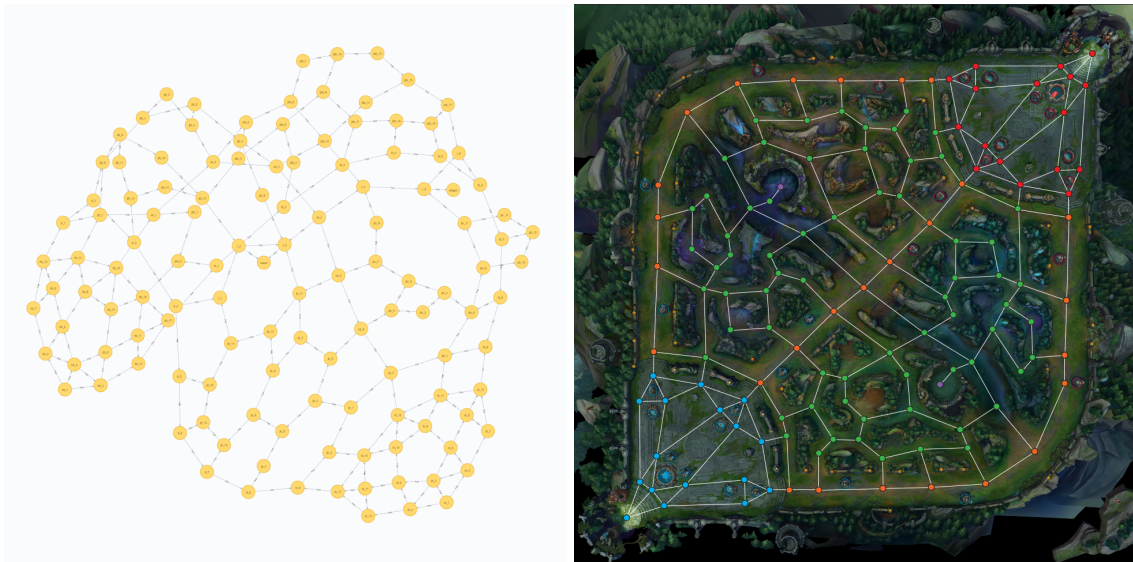


Figure 3.4: Neo4j nodes graph (left) and nodes represented in Map (right)

### CouchDB

While handling requests to the API when searching for player matches (Figure 3.1), the requests would take some time given the restriction imposed by the API (Subsection 2.5.2). One possible solution was to add database support that would save the search content as a JSON file and retrieve that content with the following requests.

This lowered the number of requests made to the API while improving the search speed significantly. CouchDB was the elected database because it speaks JSON natively and saves information without a specific structure<sup>4</sup>, which in this case was not needed. With every request, the server checks if the user already has content in the database and if the content was updated within 30 minutes from the request time. When one of these two conditions fail, a request to the API is made instead.

Listing 3.1 shows a piece of a JSON's file representing a player's recent matches' list. This is how the information shown in Figure 3.1 is saved for later reference.

---

<sup>4</sup><http://couchdb.apache.org>

Listing 3.1: JSON file of recent matches' list

```
1 {
2   "_id": "tiagomoucho3matches1",
3   "_rev": "75-5ece340eb716752e2a545c75b69ffd68",
4   "matches": [
5     {
6       "gameId": 3689973045,
7       "gameMode": "CLASSIC",
8       "queueType": "Normal (Draft)",
9       "mapName": "Summoner's Rift",
10      "championId": 86,
11      "champion": "Garen.png",
12      "lane": "TOP",
13      "timestamp": 1530553095734,
14      "result": "DEFEAT",
15      "kills": 2,
16      "deaths": 7,
17      "assists": 1,
18      "gold": 5647,
19      "minions": 89,
20      "championLevel": 11,
21      "duration": 1432
22    },
23    ...
24  ],
25  "timestamp": 1530648221282
26 }
```

## Materialize

The design process of a web application is an important factor overall. Users like an efficient application while also being minimalist and with good aesthetics. Materialize<sup>5</sup> is a CSS framework that speeds up development and it is easy to work with. It focuses on a modern responsive front-end framework based on Material Design<sup>6</sup> guidelines. These were the reasons it was chosen to be the CSS framework used.

## Leaflet

The primary visualization technique is the animated map. To create this visualization, an open-source JS library for mobile-friendly interactive maps, Leaflet<sup>7</sup>, was used. Offering plugin support, it helps with the implementation of certain features in the animated map. Two plugins were used, *MovingMarkers* and *MiniMap*, being detailed in Section 3.5. Not using a D3.js library to power the animated map was a personal choice, avoiding the steep learning curve it requires to implement the already planned features.

---

<sup>5</sup><http://materializecss.com>

<sup>6</sup><https://material.io/guidelines/>

<sup>7</sup><http://leafletjs.com>

There are other libraries that could be used instead of Leaflet when building a geographical map lookalike. In this case, the focus should be on a custom non-geographical map, which the Leaflet library supports, while others do not. Some pre-built features also helped choosing Leaflet above others —zoom interaction is a pre-built feature that solves the problem of watching overlapping events in the animated map (Subsection 2.4).

### noUiSlider

The animated maps visualization technique is normally controlled by a time slider. The noUiSlider<sup>8</sup> is a lightweight JS range slider library with touch support and other features. It is quite customizable and comes packed with listeners, making it easier to handle and turn into a timeline. This timeline would control everything related to the visualization.

### ChartJS

In the previous prototype, static data was present, but was not enough. Users said the application should have more types of static data, which complements the primary visualization, and different visualizations to represent this data (Section 2.4). ChartJS<sup>9</sup> is a JS charting library. It offers a variety of different charts, being possible to mix them. It is totally customizable, from colours to animations, which helped represent the static data in VisuaLeague II, discussed in Section 3.5.

## 3.4 Data Processing

Riot Games' API offers the possibility to choose the data's variables, making it easier to handle the retrieved data. Although they provide some filters, the data is incomplete. Information concerning vision wards, champions and towers' health, are missing; the timeline endpoint only offers data each minute or in certain events. To simulate a pathing and events system, the data needed to be processed and, in a way, completed.

### Routing

Besides the defined nodes based on possible player direction changes, additional ones were created to smoothen the path out. Each node is connected at least to another node by a relationship named *Path*. This relationship has a *distance* attribute, resulting from the distance between two nodes. The distance between two points formula is given by:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

---

<sup>8</sup><https://refreshless.com/nouislider/>

<sup>9</sup><https://www.chartjs.org>

A Java program was built to measure the distance between all nodes using the previous formula, so it could be added to the relationship's attribute accordingly.

In the first attempt to use Neo4j as a path finder, a built-in shortest path function was used. This function returned indeed the shortest path, but the one with fewer node count. Players usually need to move fast from a point to another, except when they are trying to play safe. This ideology was put in place when deciding how the shortest path would be created. Quickly this built-in function proven to be unsuccessful, because some nodes are further away from each other, which would lead the generated path to be longer than it was supposed to, not representing the actual path made.

Given this situation, two plugins were used to make the path closest to its actual representation, Neo4j Spatial and APOC. The Spatial library enables the use of spatial operations on data, like searching for data within a region or within a specified distance of a point. The first step was to create a point layer where later the nodes will be referenced to. Using the Cypher language, the nodes were created with unique id's for better readability, and more two parameters representing both latitude and longitude, required by the library. After being created, they are simply added to the point layer and the available spatial functions can be used. There was only one function used in this case, the closest function, which finds all nodes in the layer within the distance to the given coordinate. This function is important to know what is the first node the player must go to. The APOC library consists of many procedures to help in areas like data integration, graph algorithms or data conversion. With this library, the Dijkstra's algorithm was used to find the shortest path between nodes based on their distance. Using this function, instead of the previous one, the path is now actually the shortest between nodes.

With the player's coordinates at a timestamp, we first find the closest node to his coordinates. The coordinates of the following timestamp are also used to find the closest node, establishing the path's start and end nodes. Making a call to Neo4j with these nodes using the Dijkstra's function, it will return the shortest path between them, with information of every node the player must move to in succession, avoiding going through walls.

With the paths being created correctly, the next step was to calculate the movement's duration between each nodes (Listing 3.2). To find speed, we need to divide distance by time. Dijkstra's function not only returns the path, but also the total distance. This total distance is incomplete, though. We also have to consider the distance between the point where the champion starts and goes to the first node, and the distance between the final node and the point where the champion must be (lines 3–5). So, dividing total distance by the subtract between the end timestamp and start timestamp, we get the constant speed a champion must travel through the route (line 7), mimicking LoL's gameplay.

Having the speed constant, we loop through each route's node. To calculate the duration between the current node and the next, we need to divide the distance by speed,

adding it to the duration's array, not forgetting the first and last points that need to be considered (lines 9–15). Finally, the route must be added to the path array (line 16). Both arrays will then be used by the MovingMarkers plugin, simulating movement.

There is the possibility that the start and end nodes are the same, which will make the Dijkstra's function to not return a route. The path will then be directly between the start and end nodes, and the duration will be the subtract between the start and end timestamps (line 18). HTTP requests are being made to Neo4j to create the paths for all players when a match is being loaded.

Listing 3.2: Prepare route function's callback

```

1  model.prepareRoute(start, end, function (route) {
2      if route exists {
3          Calculate distance between start point and first route node
4          Calculate distance between last route node and end point
5          Sum both distances to the total distance returned in the route
           ↪ variable
6
7          Calculate speed = total distance / (end timestamp - start timestamp)
8
9          Add the first duration to the array calculating first distance /
           ↪ speed
10
11         for route between each node {
12             Add duration to the array calculating route distance / speed
13         }
14
15         Add the last duration to the array calculating last distance / speed
16         Concat the route to the path array
17     } else {
18         Add duration to the array with (end timestamp - start timestamp)
19     }
20     Callback
21 });

```

## Match Data

For each match's participant, the API gives information about the summoner, the played role and lane, the runes used, among others. Listing 3.3 shows a portion of already processed data sent to the front-end.

Inside the *match* key we have 5 important pieces of information: *matchData*, *gameDuration*, *items*, *runes* and *version*. The *matchData* includes all 10 participants, identified by their id's. Inside each participant's information, we have 2 sections: *userInfo* and *championInfo*. The first one comprises information about the summoner. The latter provides information about the specific champion the summoner played. The *matchData* represents part of the static data shown in the information panel. The *items* and *runes* keys are



lists of every items and runes, respectively, present in game. These values are accessed by the information panel when building runes' list of every participant, and items' list updates throughout the game, discussed in Section 3.5. The *version* is used in the Uniform Resource Locator (URL) Data Dragon server, provided by Riot Games, to request the latest images.

Listing 3.3: Match Data JSON content example

```
1  {
2    "match": {
3      "matchData": {
4        "0": {
5          "userInfo": {
6            "summonerName": "tiagomoucho3",
7            "participantId": 3,
8            "lane": "Bottom",
9            "role": "DUO_SUPPORT",
10           "runes": [...]
11         },
12         "championInfo": {
13           "id": 89,
14           "key": "Leona",
15           "name": "Leona",
16           "title": "the Radiant Dawn",
17           "image": {...},
18           "tags": [...],
19           "info": [...],
20           "spells": [...],
21           "passive": {...}
22         }
23       },
24       ...+9
25     },
26     "gameDuration": 1805,
27     "items": {...},
28     "runes": {...},
29     "version": "8.12.1"
30   },
31   "timeline": {...}
32 }
```

## Timeline Data

Being Node.js asynchronous, the Timeline Data was one of the most difficult to build. We need to consider information each minute, alongside events that may happen between minutes. However, they need to be time-ordered which goes against the asynchronous methodology. Listing 3.4 shows an example of how the data is formatted after being processed throughout the raw JSON data.

Listing 3.4: Timeline Data JSON content example

```

1  {
2    "match": {...},
3    "timeline": {
4      "timelineData": {
5        "participantFinal": {
6          "0": {
7            ...,
8            "60000": {
9              "path": [...],
10             "duration": [...],
11             "status": "ALIVE",
12             "level": 1,
13             "totalGold": 500,
14             "currentGold": 0,
15             "xp": 0,
16             "minionsKilled": 0,
17             "kills": 0,
18             "deaths": 0,
19             "assists": 0,
20             "items": [...],
21             "q": 1,
22             "w": 0,
23             "e": 0,
24             "r": 0
25           },
26           ...
27         },
28         ...+9
29       }
30     },
31     "eventsJSON": {
32       "3000": [
33         {
34           "type": "ITEM_PURCHASED",
35           "itemId": 1055,
36           "participantId": 10
37         }
38       ],
39       ...
40     },
41     "teamStats": {
42       ...,
43       "180000": {
44         "blue": {
45           "baron": 0,
46           "dragon": 0,
47           "riftHerald": 0,
48           "inhibitors": 0,
49           "towers": 0,
50           "gold": 4693
51         },
52         "red": {
53           "baron": 0,
54           "dragon": 0,
55           "riftHerald": 0,
56           "inhibitors": 0,
57           "towers": 0,
58           "gold": 4076
59         },
60         "towersArray": []
61       },
62       ...
63     }
64   }
65 }

```

We have 3 different sections here: the *participantFinal*, containing each participant and a list of timestamps with information inside each one of them. The *eventsJSON*, used to update the visualization status as the game plays and used in the notification system, discussed in the Section 3.5 ahead. And finally, the *teamStats*, containing information about the teams, like team gold, monsters defeated and structures destroyed. All 3 sections are built with a timestamp structure within, defining the order when the paths, durations, events and stats should update. This structure is in milliseconds as the raw data is handled in milliseconds as well.

In the participants' list, inside each timestamp that equals an exact minute, information about the champion's stats is saved like current level, Kills, Deaths, Assists (KDA), abilities, among others. In other timestamps, only path, duration and status variables are present. This happens because Riot Games' API only gives certain information with each

minute. Again for team stats, the information is only available with each minute, so the timestamps will only be exact minutes of the game's total duration.

To ease the process of asynchronous programming, the Async<sup>10</sup> library was used. This library helps encapsulate code so it can be run "sequentially". When a portion of the code finishes, a callback is used to let the process know that it is ready to go to the next portion. For better understanding, Listing 3.5 presents the pseudo-code written sequentially.

Although being presented as sequential code, the code sequences are actually encapsulated in one another, using callbacks to check when the sequence had run its course, mostly the *for each* cycles. Taking this into account, the first step was to initialize the participant's data in each minute. Since the given coordinates are part of a greater resolution than the map's image used by Leaflet, it was necessary to transform them to fit the lower resolution. A function to transform the coordinates was applied to create the now corrected coordinates (lines 7–11). Meanwhile, for each minute, events were filtered by its type using a switch statement (refer to Table 2.3). Events like *WARD\_PLACED* and *WARD\_KILL* were not used since they do not give spatial information. A custom event was created, *CHAMPION\_ALIVE*, to let the timeline know when a champion can be shown again, also known as respawn time, discussed in the following Section 3.5. The champion's respawn time is calculated through a formula that takes into consideration the champion's level and the game's current minutes. The bigger these variables, the longer the respawn time will be. All the information regarding events is done between lines 14–19. In the initial minute (zero), however, events are nonexistent. So, we skip this minute frame when checking for events (line 25).

Given the asynchronicity, a library was used, *sort-by*<sup>11</sup>, to help order the data by its timestamp (lines 30–33). Another library was used, *lodash*<sup>12</sup>, to help deal with arrays and strings.

When the data is finally ordered, the next step was to build the paths for each champion, making calls to the Neo4j database in the *prepareRoute* function (line 37), using the functions mentioned in Section 3.4. As soon as the path is ready, a callback is sent to let the process know it is ready to continue (refer to Listing 3.2).

---

<sup>10</sup><https://caolan.github.io/async/>

<sup>11</sup><https://github.com/kvneff/sort-by>

<sup>12</sup><https://lodash.com>

Listing 3.5: Prepare Timeline Data function

```
1  let prepareTimelineData = function(rawData, callback) {
2      Initialize participants list
3      Initialize auxiliar variables
4
5      // Initialize and populate all data structure
6      for each minute frames in raw data {
7          for each participant in raw data {
8              Initialize content of participants list in current minute
9              Transform participant coordinates
10             Add coordinates to the current minute participants list path
11         }
12
13         if minute frame is not zero {
14             for each event in raw data {
15                 Switch between event types
16                 Add position to the participants list path in the current
17                     ↪ event timestamp
18                 Add event to events JSON
19                 Add and/or update content to auxiliar variables
20             }
21
22             for each participant in participants list {
23                 Add content of auxiliar variables to the participants list in
24                     ↪ current minute
25             }
26         } else {
27             Next minute frame in raw data
28         }
29     }
30
31     // Need to order all data in chronological order before continue
32     for each participant in participants list {
33         Order data by its timestamp
34         Change champion status (alive or dead) in the timestamp accordingly
35     }
36
37     // With the ordered data, we can now build the participants paths
38     for each participant in participants list {
39         Build paths using prepareRoute function
40     }
41
42     Return processed data to the front-end
43 }
```

### 3.5 Prototype Features

Some of the VisuaLeague II implemented features are the result of solutions found to mitigate the previous prototype limitations. Others are simply required to achieve the animated maps visualization technique. Extra features were added to improve the user experience and aid in the analysis. These features are shown in an expanded view of the application in Figure 3.5, and are broken down in this section.

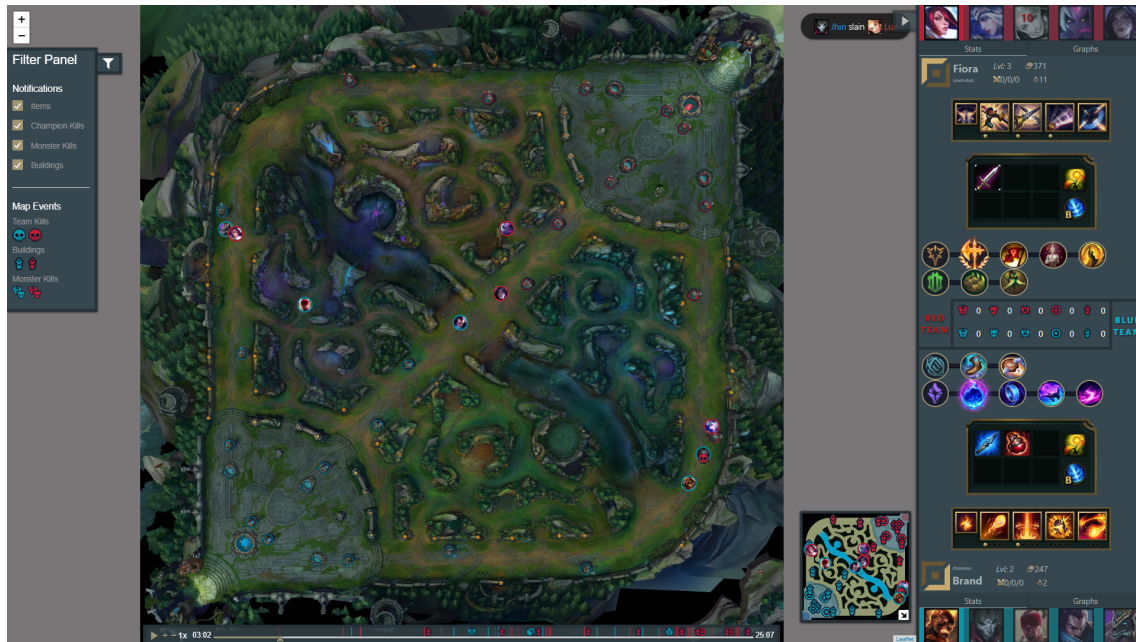


Figure 3.5: VisuaLeague II application expanded

#### Map Visualization

LoL's map does not have a top-down perspective, neither it supports zooming out to a complete wider map's view. The players are limited to a specific window, approximately close to the champions. In order to build VisuaLeague II map representation, a tool was used to change the camera's perspective. Normally, SkinSpotlights Creator Suite<sup>13</sup> is used to get different perspectives of a champion's skin in-game, but in this case it was used to get a map's top-down view perspective. A custom game was loaded and a screenshot was taken with maximum resolution possible (4K) to give a better detailed view when zooming in.

Having the map's image ready, the next step was to initialize the map's representation using Leaflet. As the map's image has margins, it was necessary to define the map bounds so the coordinates system would work correctly. It is also in this part the champions' and towers' markers are initialized, both in the main map and the minimap, with the correct representations. The minimap is another feature and is discussed ahead.

<sup>13</sup><https://www.skinnerpotlights.com/p/creator-suite.html>

The main map has already defined towers because they are already present in the map's image, so we needed to find a solution to indicate the tower status, active or destroyed. In the minimap this was easier to do, because the image is at a clean state (no structures defined), so the towers locations were added as markers and could be removed just by deleting them. The solution to represent the towers' status in the main map was to add ring markers around each tower. If the ring has colour to it, the tower is active. If the tower was destroyed, the ring turns grey (Figure 3.6).



Figure 3.6: Active tower (left) and destroyed tower (right)

The markers can be initialized with tooltips. These tooltips can provide additional information about a specific marker, for example, information about a kill if hovering over a death marker, or a champion's and summoner's name if hovering over a champion's marker.



Figure 3.7: Skull shaped marker and its tooltip representing a champion kill

These death markers represent kill events. When a champion dies, a skull shaped death marker is placed where the kill happened on the map (Figure 3.7). Meanwhile, the champion who died is temporally hidden, until he respawns again at the right base. Once the champion respawns, the death marker is removed from the map.

Every changes to markers, being their movement, showing or hiding them, are also depicted in the minimap, except for the tower rings, which their representation is different. The tooltips, however, are only displayed in the main map.

## Timeline

The animated maps visualization technique requires a time controller, being able to go forth or back in time. VisuaLeague II timeline is a slider where the user sees time progression, the current time and a match's duration. Above this slider is represented an events' timeline, showing where match events occur in the space of time. The user also has buttons to control the time and time's speed, up to 3 times.

Figure 3.8 portrays the timeline's design evolution. These design decisions were guided by LoL players. Initially the timeline was too tall, overlapping the main view a bit. It started to get thinner and wider, so the now added events' timeline could be depicted without much events overlapping. Finally, the death's icons were replaced by lines, resulting in a better timeline's visualization.

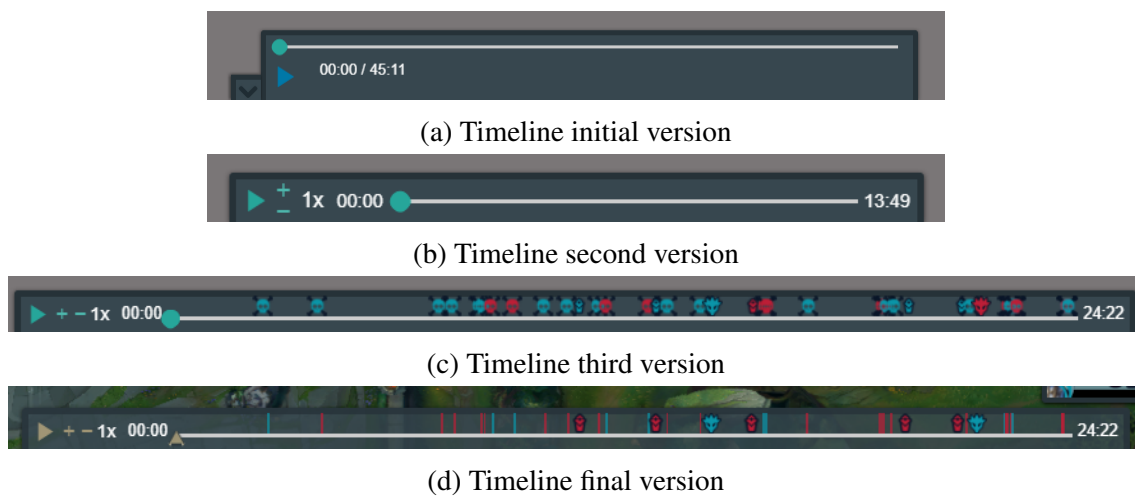


Figure 3.8: Timeline's design evolution

For every second tick, a *noUiSlider* listener event would fire checking for events in the current time at *eventsJSON*. If there are any events to be shown, it would trigger functions to show notifications, new markers (a skull indicating a champion's death) or update stats in the informations panel. Dragging the time handler shows only exact minutes the player can go to. When the handler is dropped, not only the *eventsJSON* is checked, but also the *teamStats* and participants' level, KDA, updating as necessary. This update also happens when the time reaches an exact minute.

Basically, the timeline's listener events fire up functions which will change the entire state of the visualization, from markers being displayed or hidden on the map, to stats being updated in the information panel. If the user goes back to a previous timestamp, it is ensured that events that should have not happened yet are taken out and events that already happened and should be shown are indeed shown.

To optimize the calculations, updates and state changes, information relative to every minute the user had been to was saved in the browser session storage, so that it should be



loaded up immediately and bypass all the previous necessary calculations whenever the user went to the same minute again.

### Markers Movement

Having the paths and paths' duration arrays it is essential to initialize the champion's markers so they can move. The *MovingMarkers* plugin helps creating markers' movement, given an array of positions (path) and an array of durations as arguments. The durations array help define how long the marker will move between two positions. This plugin calculates the interpolation between two points, thus creating the motion.

It comes with pre-defined functions to stop, pause and resume the markers' movement, so we only need to synchronize with the timeline's controls. It also allows to control the movement speed, so whenever the user rises or lowers the timeline's speed, the plugin will automatically calculate and change the durations' array.

### Minimap

Using another Leaflet's plugin, the *MiniMap* plugin creates a small map in the bottom right corner of the main map (Figure 3.5), which shows the same as the main map with a set zoom offset. However, for our application we wanted a shrunk view of the actual map. This was able to do by increasing the minimap's image resolution to match the one from the main map and adding similar bounds. Officially, the minimap does not provide markers representation on it, but since the plugin is a extension of a map layer, we had access to pre-built Leaflet functions to help adding markers to the minimap.

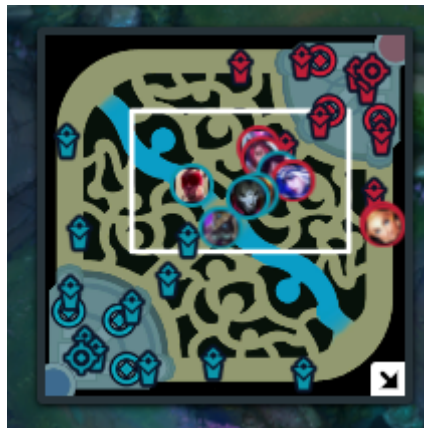


Figure 3.9: VisuaLeague's minimap

In this representation, the markers are smaller comparatively to the ones used in the main map. The structures are also shown differently, with icons changing according to the structures they represent. When a structure is destroyed, the marker is removed from the minimap and, if a champion dies, it is hidden. If the user zooms in enough, a outlined white rectangle is shown depicting the window the user is currently viewing.



The image used for the minimap is different from the main map because such a detailed view is not required, and it is similar to the minimap image used in the LoL game to be more familiar to players. As this is an analysis tool for LoL players, the images used in stats, markers and other features are similar to those used in the real game, decreasing the learning curve.

The minimap was developed to offer map's context when player's zoomed in and could not see the surroundings. This is directly tied with the mitigation of overlapping events, since zooming in was the found solution.

### Notifications

It was developed a notification system to alert users of what was happening as they watched the match. These notifications are visual representations of events, such as items purchases, champion's kills and destruction of structures (Figure 3.10).

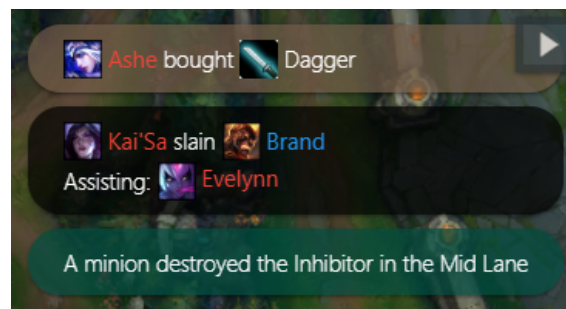


Figure 3.10: VisuaLeague II types of notifications

The notifications are colour-coded depending on the event they portrayed for users to better distinguish the events. They are located in the top right corner (Figure 3.5) and are dismissed after 4 seconds. The notifications were implemented using *Materialize's* toasts and their appearance is controlled by the timeline.

### Filter System

For users to have greater control over the data they want to see, a filter system was implemented. The panel located on the application's left side (Figure 3.5) is divided into two categories: notifications and map events. The first one filters all the information regarding notifications depending on what is selected. The second one filters the timeline's events by hiding what it is not selected. Only the team kills filter have impact on the map and minimap, not adding the skull marker when a champion dies.

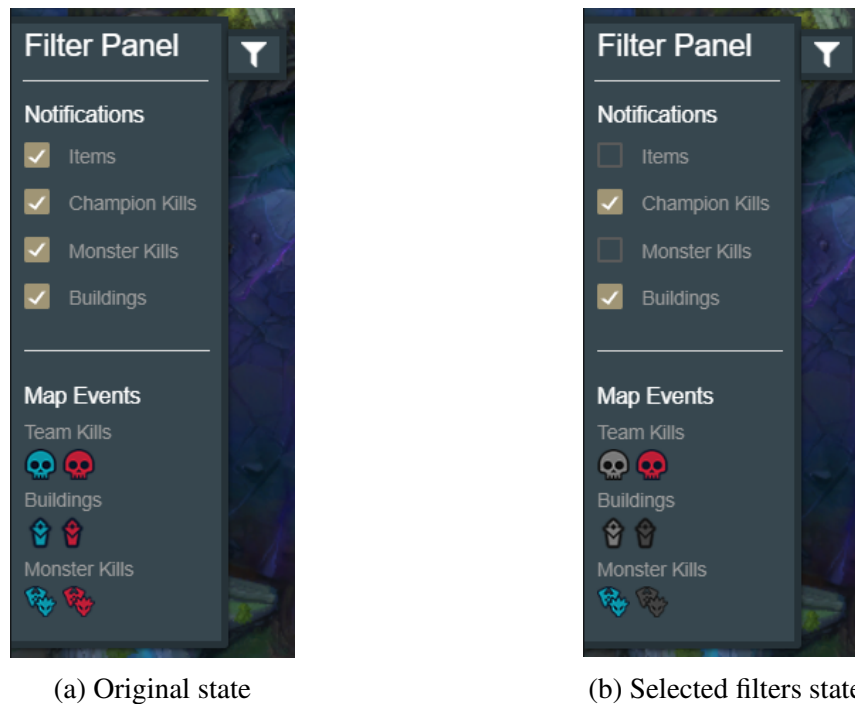


Figure 3.11: VisuaLeague II Filters Panel

Figure 3.11 shows the original state of the filters panel followed by the same filters panel but with some selections. Figure 3.12 shows the result of the selected filters in Figure 3.11b in the timeline's events.



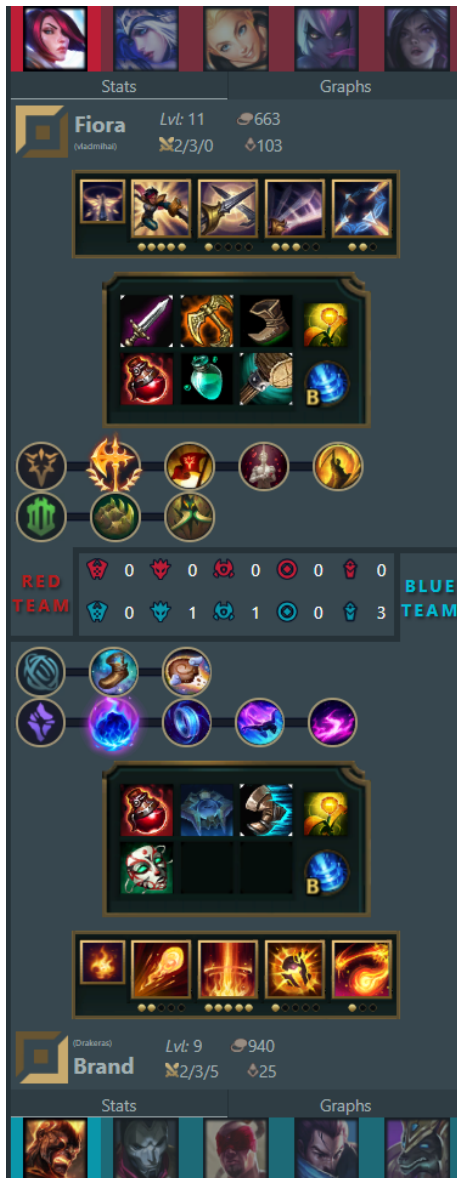
Figure 3.12: Filtered event's timeline

The filter system was initially intended to have more options, from customizing options for every champion, to being able to hide champions completely. Given time restrictions, every option in mind could not be implemented.

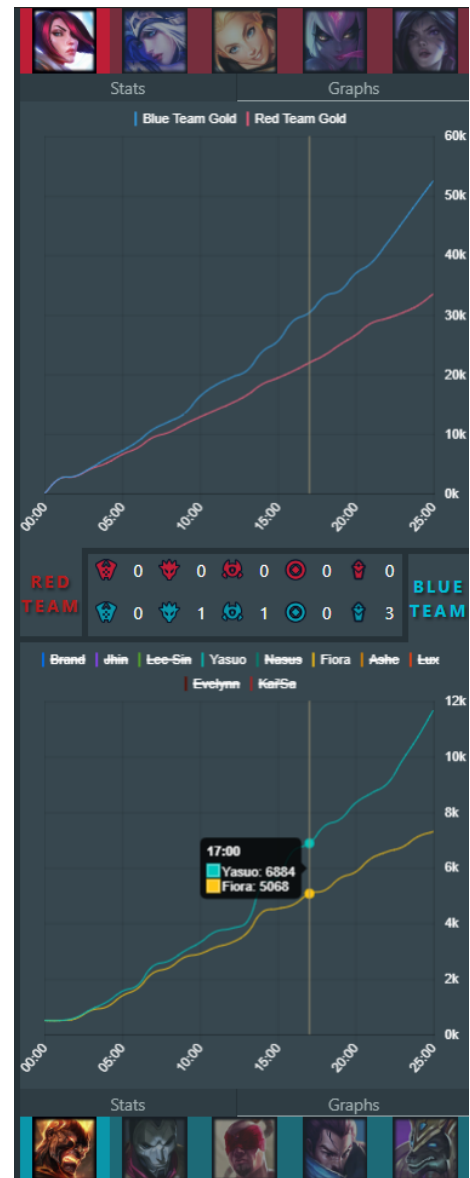
### Information Panel

Last, but not least, we have the information panel on the right side (Figure 3.5). This is where all the static and statistical data is present, from item builds, KDA, abilities and runes, to team and individual gold charts.

All the information shown in Figure 3.13 is updated dynamically as the game progresses. Figure 3.13a is the default selected panel the users first see after loading a match. The information is divided by the teams, being the red team at the top and the blue team at the bottom. The users can select the shown champion's information by clicking in their images. The current selection has a more vivid look at the champion's image, while the other images are more faded.



(a) Stats Panel



(b) Graphs Panel

Figure 3.13: VisuaLeague II Information Panel

Using the red team as reference, the champion's information is divided into four sections:

1. Summoner and champion's information — player's role, champion's name and champion's information such as current level and gold, KDA and minions killed. The summoner's name is also written below the champion's name.
2. Abilities list — this list represents the champion's passive (first icon) and the main four abilities. These abilities can be leveled up and each golden point represents a level. It is also indicated how many levels are left to maximize the ability.

3. Items build — players items' purchases and sales are represented here. When a player purchases an item, the item is placed in an available slot.
4. Runes list — these represent the runes selection players made before starting the game. This data is completely static.

The same type of champion's information is represented in the bottom for the blue team players, but in a mirrored-order. Between the adversary player's information are represented team stats. The team they belong to is shown through the icons colours. These stats tell how many structures a team has destroyed and which elite monsters they have killed.

When users select the graphs tab, the information is switched up (Figure 3.13b). The gold charts take place of the champion's information. At the top, a team gold chart is shown with red and blue lines representing each team. At the bottom, individual gold chart is shown with the possibility to add or remove lines. Each line represents a champion. Warm colours were chosen for the red team champions and cool colours for the blue team champions. When hovering a chart, a tooltip is displayed with detailed information.

For optimization purposes, after the match is loaded, the charts are fully drawn and not as the game progresses. However, a line in the x-axis is drawn to show the current minute and is updated with each minute.

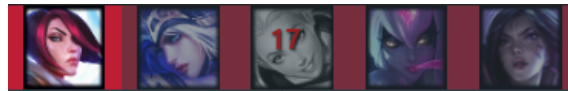


Figure 3.14: Death status in information panel

Certain events that are represented on the main map also interfere visually in the information panel. Figure 3.14 shows what happens to a champion's image when it dies. The champion's image turns black and white and the respawn timer is shown, decreasing with each second. When the timer reaches zero and the *CHAMPION\_ALIVE* event triggers with the timeline, the champion's image reverts to its original colours and the timer disappears.

## 3.6 Summary

With the developed prototype, this chapter aimed at illustrating the technologies and architectures used, the features that were implemented, describing the full process behind the development of VisuaLeague II.

First, the VisuaLeague II prototype was shown, where some advantages of building web applications were explained. The architecture was illustrated followed by a discrimination of every technologies used to built this application, from back-end tools, to front-end development.

Next, three strands of data were described, showing how they were processed and divided with code and JSON examples.

Finally, all of VisuaLeague II visualization techniques are depicted, explaining in detail how they were implemented, and how they connected and work with each other.



# Chapter 4

## Evaluation of VisuaLeague II

Next to the VisuaLeague II development, a user study was conducted to assess the application as an analysis tool, using animated maps. For this purpose we have conducted a comparative study between VisuaLeague II, OP.GG and LoL's Replay to evaluate which tool is more adequate to analyze player's performance.

This chapter presents the process conducted to evaluate VisuaLeague II. The evaluation was conducted in two phases: informal interviews and a comparative study. Section 4.1 presents the informal interviews that served as an application features test. Section 4.2 describes the applications used in the comparative study and in Section 4.3 the conducted comparative study, followed by the results obtained in Section 4.4. Finally, a discussion of the results is present in Section 4.5.

### 4.1 Informal Interviews

At the Dia Aberto<sup>1</sup> event at Faculdade de Ciências da Universidade de Lisboa, VisuaLeague II was made available to many visitors that could freely interact with the application. The main goals were for bug testing, design changes and overall interest. Several high school groups used VisuaLeague II and expressed their suggestions, saved for further analysis. It is important to mention the application was not yet finished.

Before letting the visitors test the application, a brief introduction explained the concept of VisuaLeague II and the importance of using animated maps to analyze spatio-temporal data. They showed interest in using VisuaLeague II because of the possibility of reviewing a match without the need of additional software. It was also different from what they usually use (OP.GG), as they could see the players' movement through the map.

Many of the visitors expressed concern at the lack of statistical data. At this point, statistical data was not yet implemented, but suggestions were taken for future implementation. Although they could see more than what they are used to with the other tool, visitors said they would use statistical data to better understand what is happening in the

---

<sup>1</sup><https://ciencias.ulisboa.pt/pt/dia-aberto>

primary visualization.

## 4.2 Considered Applications for Comparison

The analysis of LoL matches’ is not something new. Users have tools they can use, gaining access to matches they, or others, have played. These tools provide general game statistics referring to the players’ performance or a gaming session. Two applications (OP.GG and LoL’s Replay System) are going to be used to assess VisuaLeague II as an analysis tool, since the first is one of the most commonly used third-party application tools and the latter an official system to review matches, both being incredibly popular.

### 4.2.1 OP.GG

This web application uses visualization techniques such as diagrams, charts and plots to show a summary of statistical data regarding player’s performance after a match is over, such as total gold earned, item builds, KDA. Figure 4.1 shows a summoner’s initial page with a match currently selected. The user can check finals stats for each player, but only has access to the current summoner’s item build and abilities progression. If the user wants to check other player’s items and abilities progression, new tabs must be opened.



Figure 4.1: OP.GG Summoner Page with match view

Statistical data is the strength of OP.GG, having different types of charts depicting team gold, advantage in kills, experience over time, among others (Figure 4.2). Finally we have a map where kills are represented, and a textual timeline (Figure 4.3). Every



kills/deaths are marked on the map, with the possibility of showing the affected players and when it happened. However, the user must go through every marker to identify the players and the timestamp. There is also a textual timeline listing the match's events. The user can see what and when it happened, but does not have spatial information. The two views are shown separately, forcing the user to switch between them for spatio-temporal data.

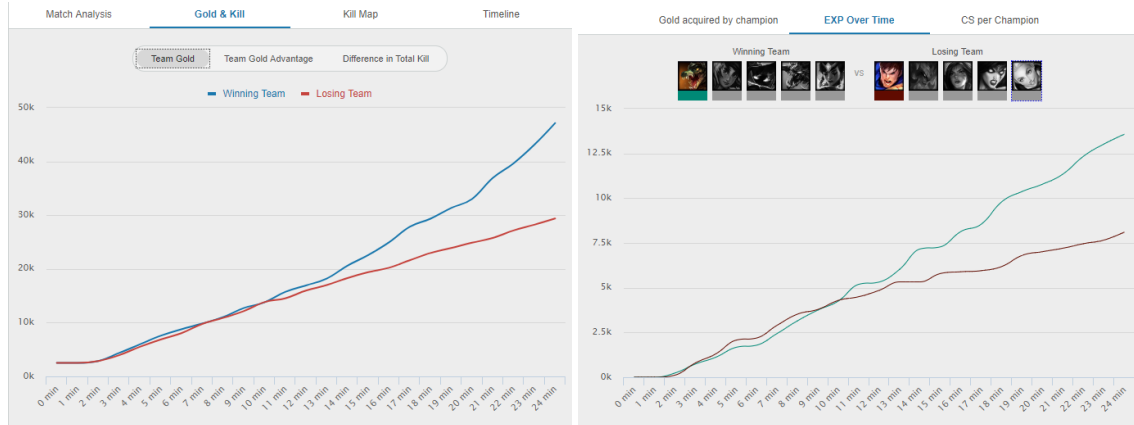


Figure 4.2: OP.GG's team gold and experience charts

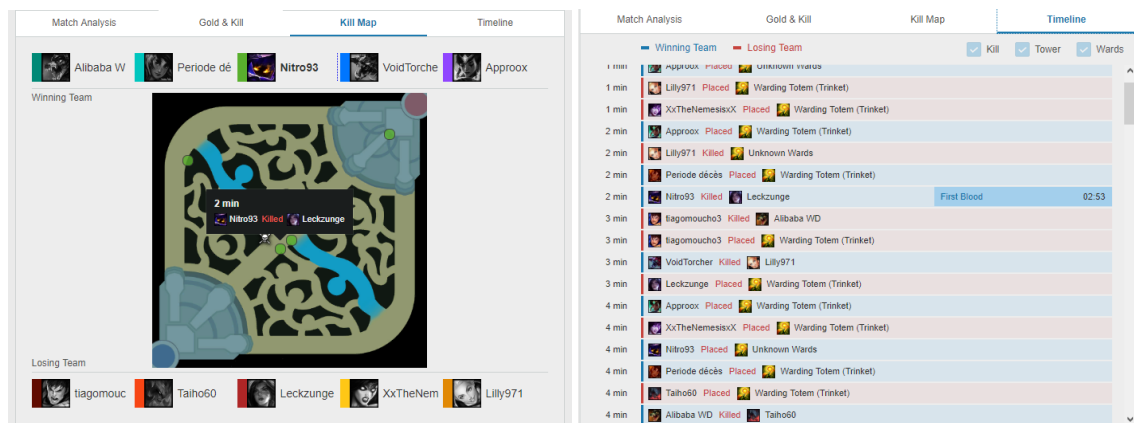


Figure 4.3: OP.GG's kill map and textual timeline

## 4.2.2 League of Legends Replay System

The replay system was developed by Riot Games, allowing playback of player's matches (Figure 4.4). This system allows jumping through time, and seeing player's information such as items, abilities and positions.

The visualization is similar to the actual game, using the same features, like the minimap at the bottom right corner. The timeline is located at the bottom middle, paired with time controls and an events' timeline. This is also where player builds and KDA's are present. Other stats, such team kills and towers destruction are shown in the top middle.



Figure 4.4: LoL's Replay System

Although its visualization is the same as what happened in the match, there are some drawbacks. Replays are only available till a new game patch, meaning they will expire as soon the user updates the client. The number of games is limited, saving only up to 20 games. The user cannot review other players' games, only the games they play. Using the replay system requires the installation of the game client and compatible hardware, not allowing access from different devices or platforms.

These restrictions have a negative impact in users trying to analyze player's performance, while wanting to see their games or other players. As far as existing applications go, this is the only one that lets players visualize and truly interact with spatio-temporal data, mainly the visualization of players' trajectories. However, this analysis can be very time consuming and needs specific software and powerful hardware.

### 4.2.3 Comparison of VisuaLeague II with OP.GG and LoL's Replay

The developed application combines components present in OP.GG and LoL's Replay. Both the statistical data shown with charts and the representation and visualization of spatio-temporal data, using the animated maps technique. Table 4.1 presents the data types each application uses. OP.GG does not have spatial data because the spatial data it presents is always followed by timestamps, so they fit into the spatio-temporal data type. However, this data is incomplete in comparison with VisuaLeague II and LoL's Replay because it only shows player's deaths time and location. On the other hand, it offers statistical data which, in turn, LoL's Replay does not.

Table 4.1: Data Types Comparison

<i>Data types</i>	<i>VisuaLeague II</i>	<i>OP.GG</i>	<i>LoL's Replay</i>
<i>Static data</i>	✓	✓	✓
<i>Statistical data</i>	✓	✓	-
<i>Spatial data</i>	✓	-	✓
<i>Temporal data</i>	✓	✓	✓
<i>Spatio-temporal data</i>	✓	✓	✓

According to the features, Table 4.2 presents a comparison between the applications' features. Since OP.GG offers a greater amount of statistical data, it relies in the use of charts and filters to visualize this data. Although, speed controls and zoom interaction are completely missing as this application does not use animation when presenting data. LoL's Replay has both of these previous mentioned features, but since it does not show statistical data, features such as charts visualization and filtering are not present. However, filtering the data could help showing the user only the information he/she is interested on. All the applications shows event's timeline, stats' information and a map view, but OP.GG is the most limited, with its event's timeline being only textual and the map representing only player's deaths.

Table 4.2: Features Comparison

<i>Features</i>	<i>VisuaLeague II</i>	<i>OP.GG</i>	<i>LoL's Replay</i>
<i>Event's timeline</i>	✓	✓	✓
<i>Speed controls</i>	✓	-	✓
<i>Zoom interaction</i>	✓	-	✓
<i>Stats' information</i>	✓	✓	✓
<i>Charts</i>	✓	✓	-
<i>Map view</i>	✓	✓	✓
<i>Filters</i>	✓	✓	-

Is is intended that the incorporation of statistical data with the visualization of spatio-temporal data improves and helps in the analysis of player's performance. As we can see from the previous tables, VisuaLeague II is the only one who has all the features and data types.

### 4.3 User Study

The user study was conducted, aiming at VisuaLeague II assessment as an analysis tool, comparing with two existing others. Participants were asked to sign a data consent form before engaging in the study. The following sections present the user study methodology and apparatus, who were the participants and the tasks they performed.

### 4.3.1 Methodology and Apparatus

The application assessment took place in a room where the participants had access to a laptop and had to interact with three different applications, OP.GG, LoL Replay and VisuaLeague II. VisuaLeague II server was initialized locally. The laptop's screen resolution was at 1366x768px, which is the native support. For hardware, the laptop uses a Intel Core i7-4500U processor, Intel HD Graphics 4400 graphics card and 8Gb of RAM. Participants were placed, approximately, 50cm away from the computer and a mouse was provided for better control.

Three different matches were used, one for each application, to avoid familiarity when transitioning between them. Although being different, the matches provided were similar to each other. The observer sat next to the participant to take notes of the interaction with the application. A smartphone was used to record the participant's analysis and comments in audio format. Before starting each application's assessment, the participants were asked a few questions, such as age, type of player and whether they had already used existing tools, and then were able to use the application freely, to remember the available tools (OP.GG and LoL Replay) if they have already used them and/or to get familiarized with, if they have not, where they could ask questions and express their doubts. A fourth match was used specifically for training purposes.

After the training phase, the participants needed to perform three tasks for each application. These tasks were the same for all applications, establishing a direct comparison between them. To minimize possible learning effects, the tasks, matches, and applications were ordered following a *Latin square* model, being identified by letters or numbers (Table 4.3). All data used during the assessment is from matches the author of this thesis played, thus helping the analysis' validation.

Table 4.3: *Latin square* model example

A	G	L
G	L	A
L	A	G

(a) Matches

V	O	R
O	R	V
R	V	O

(b) Applications

1	2	3
2	3	1
3	1	2

(c) Tasks

After all tasks, the participants were asked to compare VisuaLeague II to the others applications and suggest improvements.

### 4.3.2 Participants

A total of twenty players participated in the study, four of them being professionals and sixteen casual players. Of these professionals, two were coaches/analysts and two were players. The participants' ages vary between 19 and 29 ( $\mu = 23.45$ ,  $\sigma = 2.16$ ), and only four participants were female.

According to the tools used, more players have used OP.GG than LoL’s Replay System, fifteen and twelve players, respectively (Figure 4.5).

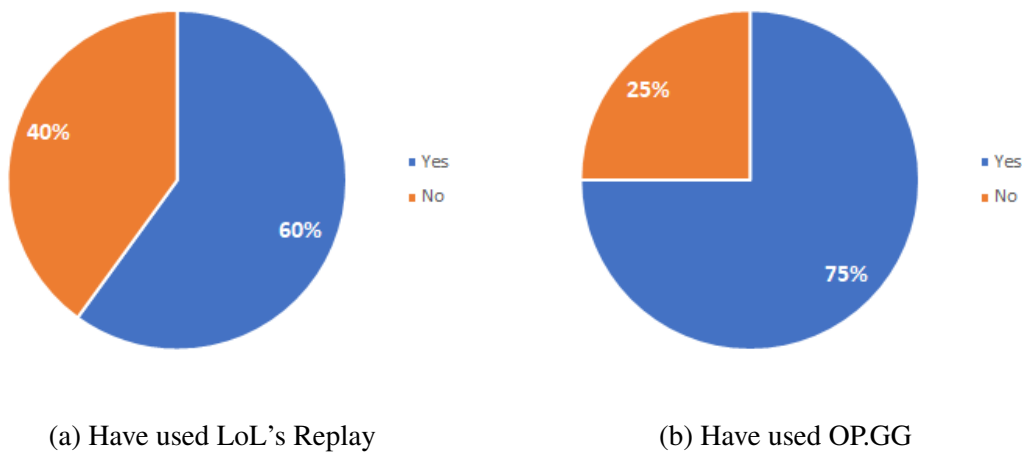


Figure 4.5: Pie charts of whether tools were used

There was an additional question for those who stated using them, where they were asked the usage’s purposes. The users could give more than one example, so the answers were coded and grouped for better understanding. For LoL’s Replay, the participants said that the purposes were to: watch key plays, watch replays, save video clips and try out only. The majority lies in watching key moments in a match, mostly direct confrontations between enemy players (Figure 4.6).

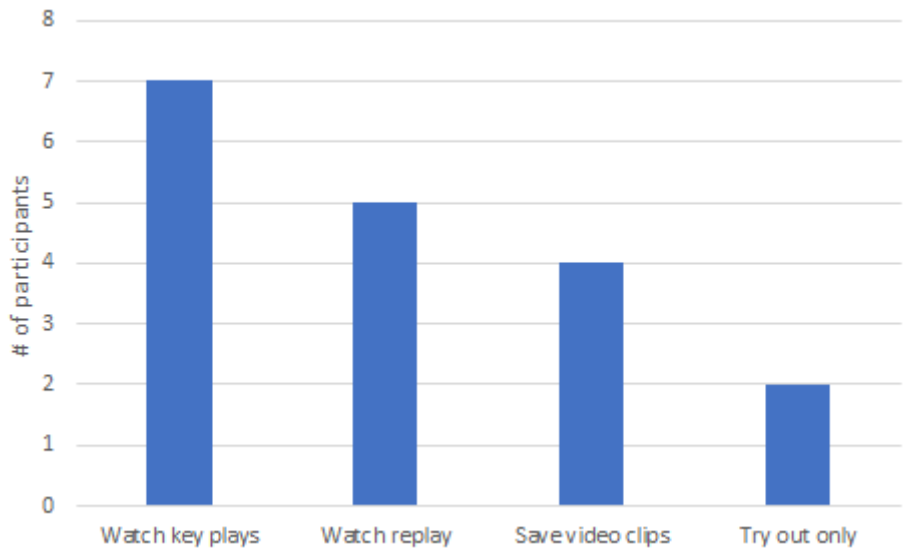


Figure 4.6: Bar chart of LoL’s Replay purpose

In OP.GG, the players said that the purposes were to: consult Match Making Ratio (MMR), check player’s information, check statistics and check item builds. Players tend

to view their MMR — which is a hidden skill rating provide by Riot Games to determine who players go against while also determining leagues and league promotions — knowing if they are above or below the average in a league; and check player's information, to know who they go up against before the match starts (Figure 4.7).

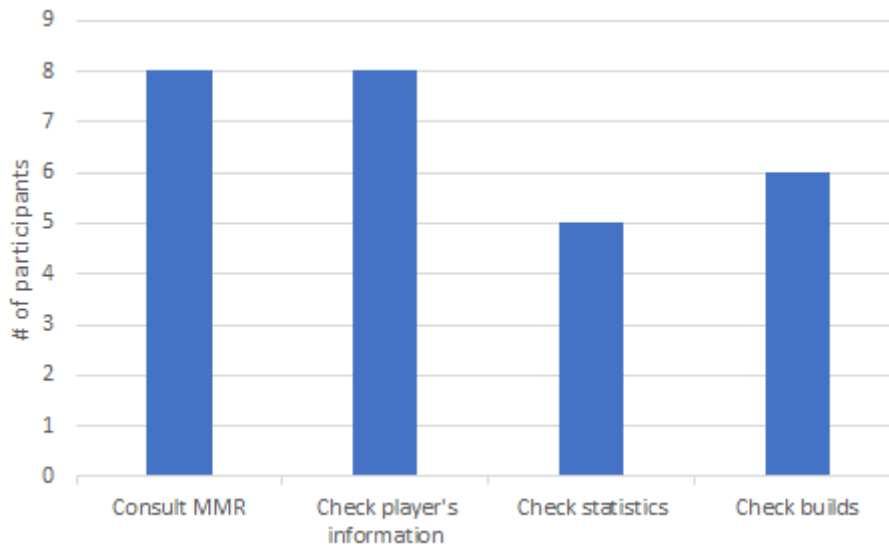


Figure 4.7: Bar chart of OP.GG purpose

### 4.3.3 Tasks

Three tasks were selected to be performed by the participants. The tasks proven to be appropriate given the analysis with three different tools. All of the tasks were the same regardless of the tool in use.

Normally, a match's gameplay can be divided into three stages: early-game (until 15 minutes), where players mostly stick to their lanes; mid-game (until 25 minutes), where the team starts grouping to help push a lane and ganking, when one or more team members invade a lane to defeat the opponent, is more frequent; and late-game (until the end), where team fights are predominant, trying to invade the enemy base and end the game.

The tasks were chosen based on the different game stages so we could get different analysis, depending on what is happening in the game, and are outlined below with a brief explanation.

#### **Task 1: Analyze the red team jungler's performance in the match's first 5 minutes.**

This task represents the early-game, where the participant must analyze the performance of the red team's jungler. The jungler tends to roam the jungle in the beginning of the game, which can be great for testing the implemented routing system. This can also help us understand what participants think is relevant and what they use to analyze a single player performance.

**Task 2: Analyze the mid lane’s gameplay from the match’s 18 to 20 minutes.**

In this task, participants were free to analyze what they think would be relevant, but were restricted to the mid lane only. With the matches chosen, from 18 to 20 minutes seemed to be a good time interval, given the fact there is always some events happening. This time interval also represents part of the mid-game phase.

**Task 3: Analyze teams performance in the match’s last 3 minutes.**

For the late-game, the participants were told to review the teams performance. They were not restricted to a lane, but in the last 3 minutes the game happens mainly in the losing team’s base. This task differs from the second one as team fights are much bigger in the latter and teams are already grouped. The analysis should focus on the team as a whole.

With each task, execution notes were taken to find out what features participants used and how. These would help us understand if the available features are sufficient, or what should be changed/added to provide a better experience.

The recorded audio was later transcribed and reviewed to see what are the most important factors in the player’s personal performance analysis, if there is any kind of pattern and what defines it.

## **4.4 Results**

With the audio portions and the textual answers, all of the participant’s answers were transcribed, coded and grouped. This section presents the results divided into three subsections: the VisuaLeague II routing system accuracy; the types of analysis the participants made for each task and the differences between tools; and the presentation of the surveys answers.

### **4.4.1 Routing System Accuracy**

The tool developed used a mixed approach of incorporating existing pathing functions with calculations of distances and speeds to generate a fully functional routing system used by the main application (Section 3.4). Since the information given to create the route is minimal, the system needed to be tested alongside the application’s assessment.

The Task 1 focus on the jungler role. Therefore, with participants analyzing the jungler’s path through the jungle, we can check the developed system’s accuracy. The actual path was marked watching the matches’ replay, defining the monsters and the order the participants need to identify. They were not told of this extra task in order to not interfere with their analysis. Although, every participant used the jungler’s path throughout the time interval to justify their performance.

Using one of the matches as example, we have the real path a jungler made in the game (the path consists in a sequence of events, e.g., the monsters the player killed, recalls to the base and ganks performed):

*Blue → Wolves → Raptors → Red → Gromp → Recall → Gank bot*

While the participants were analyzing the jungler's performance, they were stating where the jungler had been, or so they thought it had. The accuracy had two points to take into consideration:

1. The right identification of a monster's camp the jungler had been to, or plays like recalling and ganking;
2. The order in which these moments had happened.

For every correct identification of an event, a point was given. If the order was correct too, then the total would be two points. Given this example, guessing all events and their order would result in a 100% accuracy (14 points).

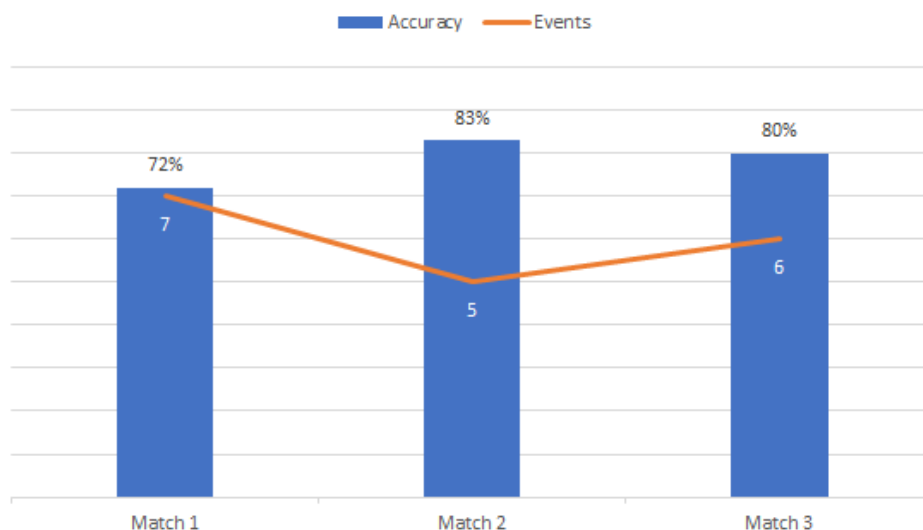


Figure 4.8: Jungler's path accuracy per match

Between the three matches, the maximum accuracy was 83% and the minimum 72% (Figure 4.8). It was possible to observe that with the increase of events to be identified, the probability of not being detected or stating the wrong order rises because many of these events are not registered by the Riot Games' API itself, which can not be shown in VisuaLeague II. Thus, the routing accuracy is going to be lower, since we only have exact players' coordinates each minute, and the participants identify these events through a player's path.

However, this sampling is not enough to be able to conclude the correlation between the numbers of events and the paths' accuracy.



### 4.4.2 Participants Analysis

During VisuaLeague II assessment, it was noted a difference in player's performance analysis made by casual players and professionals. Regardless of the tasks, professionals tend to analyze player's performance, not only telling what is happening, but also using the game's context to justify the player's actions. To them, different champions impact the gameplay since they have unique abilities. Also, characteristics like playing a AP or AD champion are factors to be considered. They would also speculate the player's decision-making and give suggestions on how he should have played.

Unlike the professionals, casual players tend to only say what is happening in the moment, using the KDA's information to justify the player's performance. However, both participant types would check and use the player's position to indicate how the player was performing. In addition, gold charts, item builds and abilities level progression were all referenced in the analysis.

The same type of analysis remained in OP.GG and LoL's Replay assessment, with professionals making a more broader and complete analysis than casual players. Nonetheless, some tools proved to be ineffective in certain tasks analysis, which are discussed ahead.

Since there were three tasks for participants to perform, the analysis was also divided accordingly to the tasks. This way, we can know what participants use for the analysis in different game situations.

#### Task 1

Indisputably, the analysis of the jungler's performance was realized mostly by the path made in VisuaLeague II and LoL's Replay. OP.GG, however, does not have information of a possible jungler's path, so the participants resorted to using charts to check the player's performance. Regardless of the used application, the participants made a direct comparison with the opponent of the same role, which means that not only they use static and statistical information to analyze the performance, but also need to make a comparison between the two players to know which one performed the better.

Every participant also tried to check the jungler's item build, if they could. With VisuaLeague II, the participants looked at the information panel to check the jungler's item build. The same happened with LoL's Replay, where they viewed the items listed. Some participants could not check the jungler's item build with OP.GG. As stated previously, OP.GG only shows build progression of one player (Subsection 4.2.1). Some participants also stated they could not make a any assumption of the jungler's performance with OP.GG, because they only had charts depicting gold or experience, which give some feedback but was not enough.

## Task 2

Although they had free will to analyze any event that happened in the mid lane, they tended, again, to compare opponents with the same role, in this case the mid laners, independently of the tool used. Even when short team fights happened, if the mid laners were present, they would mostly justify their performance, not focusing on the other team members. According to the participants, they could perceive a team fight better when using VisuaLeague II or LoL's Replay.

A direct comparison happened more frequently when using OP.GG than when using VisuaLeague II or LoL's Replay. Since they did not have a vivid representation of all that was happening at the same time, they would resort to use individual gold and experience charts to compare the mid laners. As team fights happened, the participants told which team or player was performing better from seeing the last champion standing, either with VisuaLeague or LoL's Replay.

## Task 3

Since this task specifically told participants to analyze teams performance, this was the only one where participants did not compare two players directly. They did use team gold charts only for this task. All the participants stated immediately at the beginning of the task which team was winning, mostly because, at this point, the game is at the final stage and the winning team is invading the losing team's base. Towers destruction, elite monsters killed and champions' KDA were reasons to justify why the team was winning.

In this task, however, only the professionals told right away that the game could have ended earlier if the team performed differently, for VisuaLeague II and LoL's Replay. The same could not be said using OP.GG because they don't see the champion's movement and what they are doing in real time. According to professionals, the game took longer to finish because some champions were grabbing for kills, not pushing an unprotected enemy lane, or not grouping when they had to.

## Overall

It was noted that either professionals or casual players could justify or give a more complete analysis using VisuaLeague II or LoL's Replay, due to the fact that the former has a visual representation of champion's movement and the latter being the real match. However, the analysis made with VisuaLeague II consisted of only suppositions. Either participant's types could only state for sure that the events happened or that the players were at a real position in each minute, opposed to LoL's Replay which they could clearly see what was happening.

Nevertheless, all the professionals had a more complete analysis than any of the casual players. This could be given the fact professionals have daily sessions of LoL, either

playing, analyzing or studying it. They need to have a deeper understanding of the game mechanics, and know every bit of information they can about the game so they can perform better against other professional teams, if they are players, or lead a professional team to the victory in a championship, if they are couches/analysts.

### 4.4.3 Survey Answers

Following the analysis of player's performance with each application, the participants answered a few questions regarding the analysis made, comparing VisuaLeague II directly with OP.GG and LoL's Replay. The first set of questions was about how useful VisuaLeague II was, if it was easier to use and if a faster analysis could be performed, using a 1–5 scale where 1 stands for strongly disagree and 5 stands for fully agree. The second set consisted of open answers, where participants could tell what they liked more or less about VisuaLeague II against OP.GG and LoL's Replay.

In terms of usefulness, most participants fully agreed to VisuaLeague II being more useful than OP.GG, with 60% of the votes at level 5 and 30% at level 4. Yet, most participants neither agreed or disagreed to VisuaLeague II being more useful than LoL's Replay, with 65% of the votes at level 3. Only 25% gave a 4 quotation (Figure 4.9).

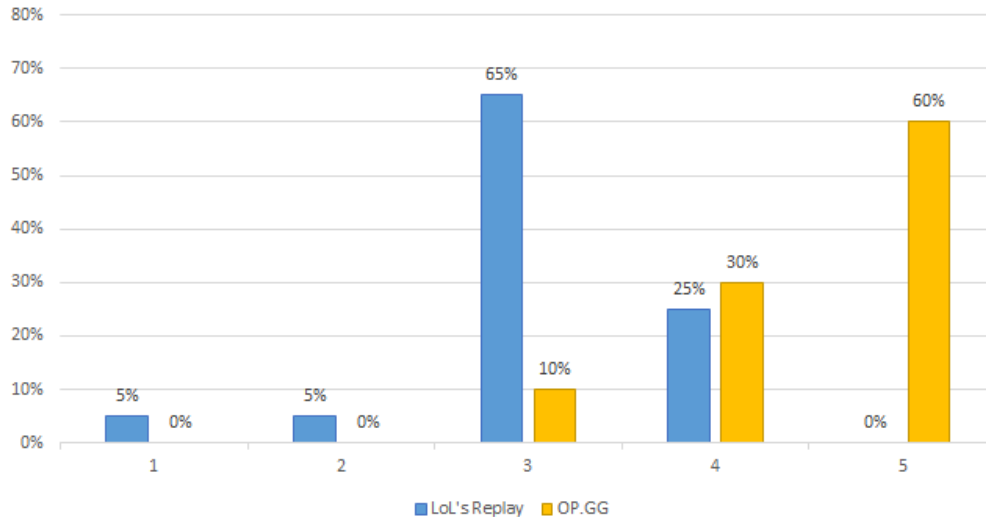


Figure 4.9: Usefulness of VisuaLeague II compared with LoL's Replay and OP.GG

In terms of easiness, most participants fully agreed to VisuaLeague II being more easier to handle than OP.GG, with 70% of the votes at level 5. There were more mixed opinions comparing with LoL's Replay. 45% of the participants agreed to VisuaLeague II being more easier to handle, but 35% neither agreed or disagreed (Figure 4.10).

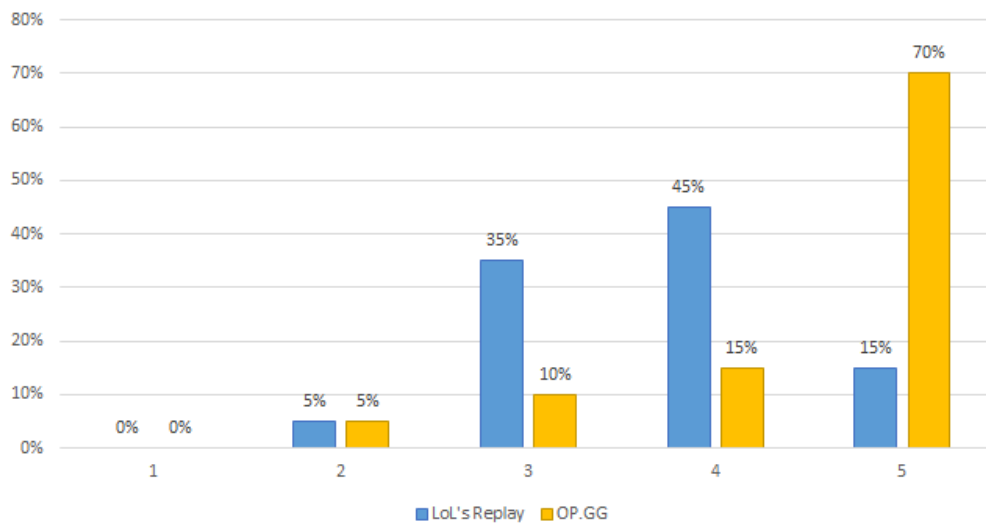


Figure 4.10: Easiness of VisuaLeague II compared with LoL's Replay and OP.GG

In terms of providing a faster analysis, half of the participants fully agreed to VisuaLeague II being able to provide a faster analysis than OP.GG. 40% simply agreed. Again, most participants neither agreed or disagreed to VisuaLeague II being able to provide a faster analysis than LoL's Replay, with 45% of the votes at level 3. 25% gave a 4 quotation and 20% gave a 5 quotation (Figure 4.11).

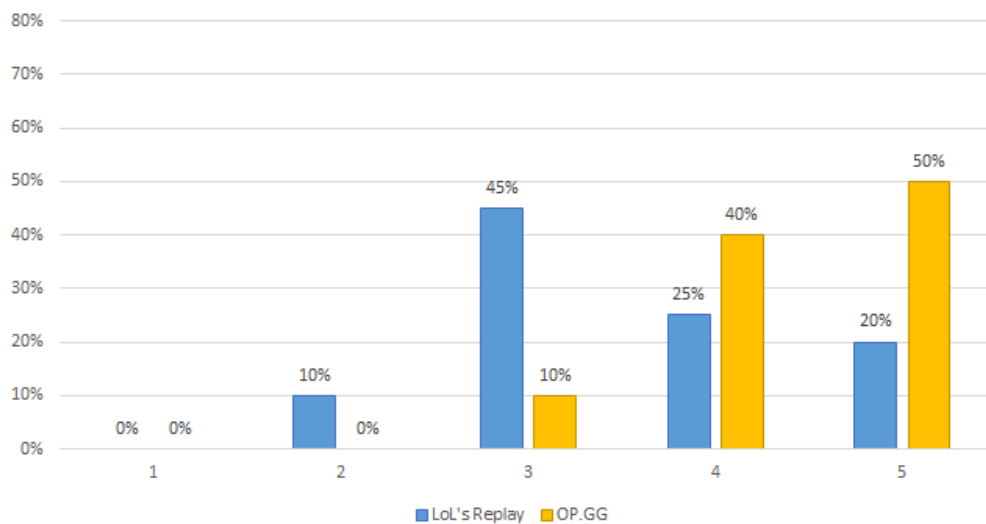


Figure 4.11: Faster analysis with VisuaLeague II compared with LoL's Replay and OP.GG

The following charts represent the second set of questions. The answers were coded and grouped for better analysis. The participants could give more than one answer, so the total may not add up to the total of participants.

Two of the most common justifications mentioned by participants about what they liked the most of VisuaLeague II in contrast to LoL's Replay were: presence of statistical data (inclusion of charts about team and individual gold); and a wider view (being able to zoom out and have an overall map's view), illustrated in Figure 4.12a. Only four participants said that VisuaLeague II has better organized information.

However, low fidelity (having only information each minute and at certain events), lack of information (such as health of champions and structures) and less defined visuals were the predominant factors about what they liked the least (Figure 4.12b).

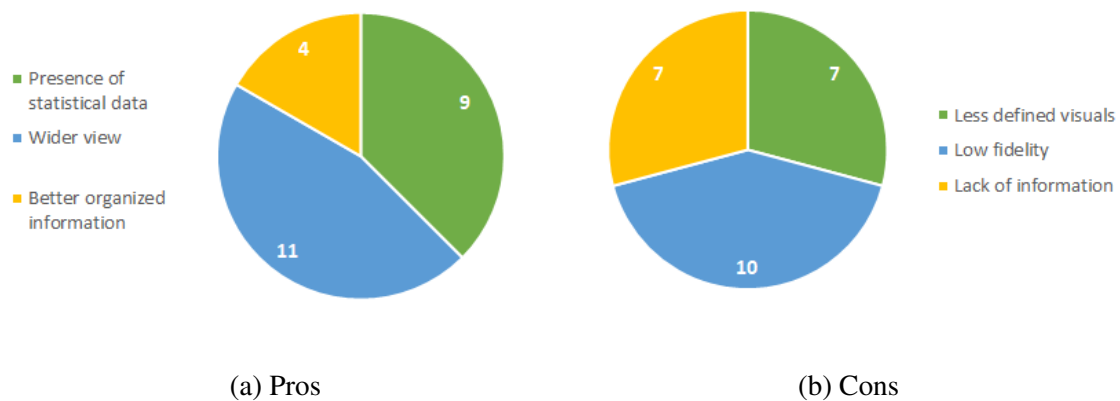


Figure 4.12: Pros and cons of VisuaLeague II against LoL's Replay

The same questions were made to compare VisuaLeague II and OP.GG. In this case, almost all participants mentioned that VisuaLeague II has a better visual representation of spatio-temporal data (usage of animated maps). They also stated that the provided statistical data is simpler and that they can have access to multiple players' information (Figure 4.13a).

The majority also said that they did not have anything they liked the least about VisuaLeague II in contrast to OP.GG (Figure 4.13b). However, some mentioned that VisuaLeague II has fewer charts (missing charts of minions kills and experience per champions) and fewer final statistic data (missing KDA ratios, and total damage dealt and taken).

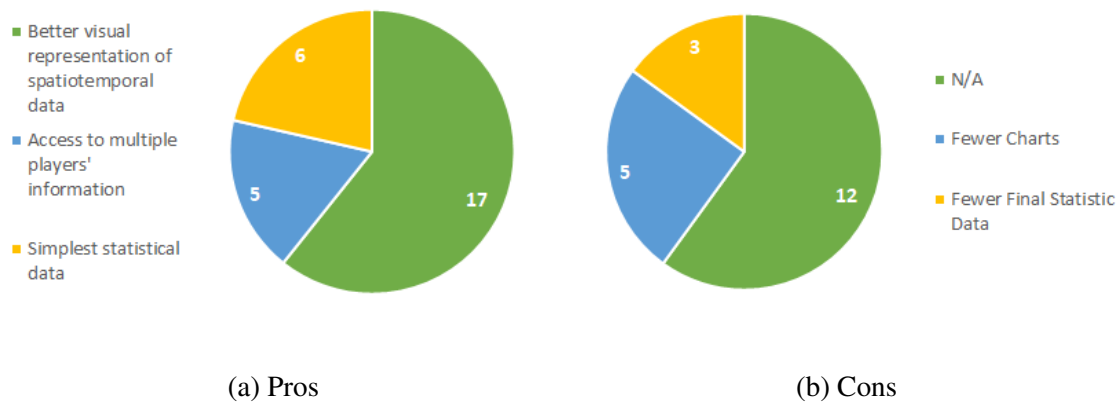


Figure 4.13: Pros and cons of VisuaLeague II against OP.GG

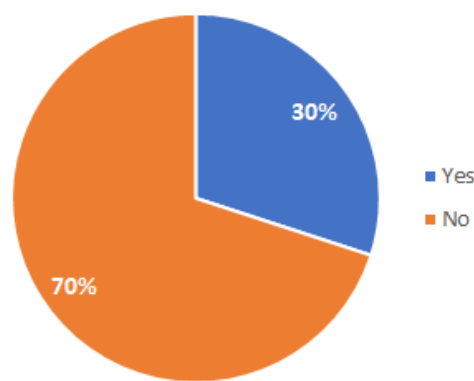


Figure 4.14: Pie chart showing whether people would use VisuaLeague II

Although most of the casual users said that they would not use VisuaLeague II (Figure 4.14), 13 out of 14 justified they do not review LoL matches. However, all of the professionals said they would use this tool. Being free, easy to use, helping in the matches' analysis, check team positioning and a fast overall data view were the given reasons.

Only one participant justified that they would not use VisuaLeague II because of missing data. Minion waves, structures' and champions' health are essential to discern player's decision-making, leading to a better analysis. This data is not present because Riot Games API does not provide it.

## 4.5 Discussion

Following the order of the results presented in Section 4.4, it was noted that despite the fact the routing system is being based on assumptions and shortest path algorithms, the participants were able to tell approximately what happened in a game, with promising results. However, it would require a greater amount of data, mostly matches, and participants to fully test routing system.

There is a huge difference in the analysis made by professionals and casual players. Although there were only 4 professionals, their analysis was much more complete in every way than the rest of the participants. They used the whole game environment to justify a player's performance, from where the player was and the positions of players around them, to the champion's type the player was using, explaining that different champion's abilities impact the game and other players differently, and so different strategies must be put in place. They have always used statistical data to justify their performance analysis, mostly gold and experience charts.

Casual players, however, simply stated what was happening in the game, such as deaths, item purchases and towers destruction. Many of them justified that one player performed better than the other solemnly for their amount of gold, builds, and kills over their opponent.

Depending on the task at hand, the participants leaned more towards one tool than the others, i.e., had a better and quicker analysis assumption. However, OP.GG was always the tool with which the participants had more difficulty in making a player's performance analysis. OP.GG is the only one lacking an animated visual representation of spatio-temporal data, indicating the usefulness of animated maps in the type of analysis that participants make.

The participants could give a more detailed analysis using VisuaLeague II or LoL's Replay than using OP.GG. Some users found annoying how the OP.GG presented the champions' death data, saying that it lead to a much more time-consuming task than it should be, ultimately neglecting the analysis.

It was noted that the presence of the static and statistical data are commonly used to justify a player's performance analysis. Most of the participants always used this type of data to complement the analysis they were performing, except when using OP.GG, where they only use this type of data. In VisuaLeague II, the participants used the charts to complement the spatio-temporal data analysis on the animated map. The same cannot be said about LoL's Replay, because it does not have statistical data.

It was often seen that most participants tend to make a direct comparison between players when making a player's performance analysis. As said above, they tend to use the amount of gold, items and kills to know if one player is ahead of the other, thus shaping their player's performance analysis.

Participants make more assumptions with VisuaLeague II than with LoL's Replay.

This is due to the lack of information that the API provides, which in turn causes the participants to not make a better player's or teams' performance analysis. With LoL's Replay the participants do not make assumptions, because they can really see what happened. VisuaLeague II does not have such a faithful representation, leading players to a more assumption analysis, which is completely understandable.

Looking to the survey's results, we can undoubtedly realize that users preferred using VisuaLeague II instead of OP.GG. However, there were some doubts when choosing between VisuaLeague II and LoL's Replay. VisuaLeague II could always, or almost, be more useful, easier to use and perform a quicker analysis than using OP.GG. This is because VisuaLeague II has a better visual representation of spatio-temporal data, namely the use of animated maps. Most participants also had nothing to say negatively about VisuaLeague II other than the lack of different charts representing statistical data.

Comparing VisuaLeague II with LoL's Replay, most participants could not tell which one was easier to use than the other. This is also true for usefulness and for a faster analysis. One of the advantages of VisuaLeague II was the possibility of viewing the map as a whole, something that LoL's Replay does not support. However, the reasons for not choosing VisuaLeague II against LoL's Replay always lie in the fact that it is less faithful, has less data or weaker visuals.

Casual players would not use VisuaLeague II. However, this is not due to VisuaLeague II being a bad tool, but rather these players do not review LoL matches. As casual players, they simply play for fun, with no intentions of improving their performance.

All the professionals said they would use VisuaLeague II and that professional teams are lacking a tool of this type. The two coaches/analysts said they spend a lot of time analyzing matches. There are several types of preparations they have to make when playing against a professional team. They build probability tables where they put champions the team may play against according to players of the opposing team and how the team's players have to behave depending on the way that an opposing player plays. There are many possibilities, strategies and decisions they have to make, and there are no free, or as practical, tools as VisuaLeague II available in the market. Most resort to watching matches' streams on YouTube<sup>2</sup>.

The study suggests that if participants had to analyze games, they would prefer using an approach similar to what VisuaLeague II and LoL's Replay offer.

---

<sup>2</sup><https://www.youtube.com>



# Chapter 5

## Conclusion and Future Work

This final chapter aggregates both conclusions and future work. In Section 5.1, a review of the present work is described, alongside with key points made during the study, referring how important the use of animated maps is when analyzing player's performance. Section 5.2 presents suggestions of improvements to VisuaLeague II or new ways to aid the player's performance analysis.

### 5.1 Conclusion

The increase in eSports popularity led different groups to be interest in online video games. The interest in analyzing data related to video games also increased, having players, coaches and analysts reviewing matches. With technologies advancements, is has become easier to extract data directly from games, using telemetry techniques to record and gather data. Telemetry data gathered can, however, be extensive, leading to a creation of big spatio-temporal data sets of information, thus proper visualization techniques must be studied and used to better analyze this data.

The work described in this dissertation focused on the redesign of the previous prototype, mitigating its limitations. Using animated maps as the primary visualization technique for the visualization of spatio-temporal data, other visualization techniques were implemented which helped the participants make better player's performance analysis. These techniques helped visualize player's trajectories, game events, and information relative to the players. The data gathered for this study is from League of Legends (LoL) matches, due to its large fanbase and longevity, providing large amounts of data ready to be analyzed, and is one of the few games that has a publicly available API that includes player's trajectories and relevant game events.

The first impressions of the informal interviews were very positive. Users showed interest in using VisuaLeague II as an analysis tool because it was different from what they are accustomed to (OP.GG), being able to analyze and make comparisons of the players' trajectories and the various events that occurred during the game. The possibility

of being able to review a game without using additional software was also a positive feature. On the other hand, users mentioned the lack of static and statistical data and how important these are to improve their analysis to better understand what was happening on the main map. Users enjoyed analyzing LoL matches through the animated maps visualization technique, but it is necessary to use more techniques to show additional information, namely the presentation of static and statistical data, to enrich their analysis.

The developed prototype was compared to two other existing tools, OP.GG and LoL's Replay. Both use different visualization techniques to visualize spatio-temporal data. The user study performed suggests that participants prefer to visualize spatio-temporal data dynamically, i.e., using animated maps. Furthermore, the visualization of static and statistical data along with the visualization of spatio-temporal data improves the player's performance analysis.

Based on the way the participants analyzed the same information and the conclusions used, they can be divided into two types: casual players; and professionals (both players and coaches/analysts). Casual players simply describe what it is happening in the visualization, e.g., *This player killed the other one and then proceeded to destroy the tower.* However, the professionals elaborate theories about a player's performance, giving suggestions on the player's decision-making, e.g., *This tower was destroyed because either the player has died many times, or she roamed. Since she only has two kills compared to the opponent who has four, this player very likely roamed more times which made her opponent push the lane and destroy the tower.* These theories, in most cases, were often coincident with the actual gameplay that took place. More Using Overall, the participants demonstrate a more positive attitude towards using VisuaLeague II and LoL's Replay instead of OP.GG as an analysis tool, because they use visualization techniques more adequate to visualize spatio-temporal data.

## 5.2 Future Work

As far as VisuaLeague II implementation goes, the addition of more types of charts would help in the player's performance analysis. The filter system could be improved in order to give more customization to players, by adding more options for each champion, instead of globally changes.

Participants suggested that the notification system could be handled differently. Sometimes they would interfere in the map's visualization, when many notifications were shown at the same time. To mitigate this issue, they could be group in a small notification panel, which could be hidden and be replaced with a bell icon, where a number would appear above this bell and increase according to new notifications being added to the hidden panel. If players want to see the notifications, they click on the bell to open the panel, and scroll through the notifications.

The routing system could be improved with more nodes' addition. Adding more nodes would mean more paths the players could walk through, meaning a possible closer representation to the players' actual pathing.

An analysis where the data interpretation is not so linear could be studied, i.e, a system where the user is able to analyze more than one match simultaneously and tell if the player's performance has improved over a number of games. This system could also help the player identify what he/she did wrong in a play or how he/she should have played. Basically, expand and be able to analyze the player's performance over several games.

Future work should also address more user studies with coaches and analysts. The results show that is an adequate analysis tool for this type of stakeholders, but it is fundamental to explore other visualization techniques to help them in the analysis and preparation of their teams and professional games.



# Appendix A

## Evaluation Form

### VisuaLeague 2.0 vs. Replay System vs. OP.GG Assessment

This survey aims to evaluate the application developed in contrast to the League of Legends' Replay system and the web application [OP.GG](#).

All data are anonymous and will be used for the purpose mentioned above. They will also be included in the final dissertation.

Thank you for your collaboration!

*\*Required*

#### Personal Information

This section serves to standardize the player type, making it easier to analyze the data.

1. Age \*

---

2. Sex \*

*Mark only one oval.*

- ☐ Male  
☐ Female

3. Type of Player \*

*Mark only one oval.*

- ☐ Casual  
☐ Professional

4. Have you used LoL's Replay System? \*

*Mark only one oval.*

- ☐ Yes  
☐ No

5. If yes, for what purposes?

---

---

---

---

---

6. Have you used [OP.GG](#)? \*

*Mark only one oval.*

- ☐ Yes  
☐ No

7. If yes, for what purposes?

VisuaLeague 2.0

The following tasks will serve to evaluate the developed application.

Player's Performance Analysis

This section consists of three tasks that will help us understand if the technique developed (animated maps) is efficient in the analysis of spatio-temporal data, the tools used and the type of analysis done.

Task 1

Analyze the red team jungler's performance in the match's first 5 minutes.

8. Execution

9. Analysis

Task 2

Analyze the mid lane's gameplay from the match's 18 to 20 minutes.

10. Execution

11. Analysis

---

---

---

---

---

**Task 3**

---

Analyze teams performance in the match's last 3 minutes.

12. Execution

---

---

---

---

---

13. Analysis

---

---

---

---

---

**LoL Replay System**

The following tasks are the same as the previous ones, but with the purpose of evaluating this tool and comparing it afterwards with the developed application.

**Player's Performance Analysis**

The tasks are the same as those carried out beforehand in order to establish a direct comparison between the developed application and the LoL's Replay system.

**Task 1**

---

Analyze the red team jungler's performance in the match's first 5 minutes.

14. Execution

---

---

---

---

---

15. Analysis

---

---

---

---

---

**Task 2**

---

Analyze the mid lane's gameplay from the match's 18 to 20 minutes.

16. Execution

---

---

---

---

---

17. Analysis

---

---

---

---

---

**Task 3**

---

Analyze teams performance in the match's last 3 minutes.

18. Execution

---

---

---

---

---

19. Analysis

---

---

---

---

---



**OP.GG**

The following tasks are the same as the previous ones, but with the purpose of evaluating this tool and comparing it afterwards with the developed application.

**Player's Performance Analysis**

The tasks are the same as those carried out beforehand in order to establish a direct comparison between the developed application and the [OP.GG](#).

**Task 1**

---

Analyze the red team jungler's performance in the match's first 5 minutes.

**20. Execution**

**21. Analysis**

**Task 2**

---

Analyze the mid lane's gameplay from the match's 18 to 20 minutes.

**22. Execution**

**23. Analysis**

Task 3

Analyze teams performance in the match's last 3 minutes.

24. Execution

25. Analysis

Final Questions

In this last phase, this set of questions will serve to have a global appreciation of the application developed in contrast to the Replay System and [OP.GG](#).

26. **VisuaLeague is more useful than the Replay System.** \*

Mark only one oval.

12345

Totally disagreeFully agree

27. **VisuaLeague is more useful than [OP.GG](#).** \*

Mark only one oval.

12345

Totally disagreeFully agree

28. **VisuaLeague is easier to use than the Replay System.** \*

Mark only one oval.

12345

Totally disagreeFully agree

29. **VisuaLeague is easier to use than [OP.GG](#).** \*

Mark only one oval.

12345

Totally disagreeFully agree

30. **VisuaLeague provides a faster data analysis than the Replay System.** \*  
*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Fully agree

31. **VisuaLeague provides a faster data analysis than [OP.GG](#).** \*  
*Mark only one oval.*

	1	2	3	4	5	
Totally disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Fully agree

32. **What did you like the most about VisuaLeague compared to the Replay System?** \*


33. **What did you like the least about VisuaLeague compared to the Replay System?** \*


34. **What did you like the most about VisuaLeague compared to [OP.GG](#)?** \*


35. **What did you like the least about VisuaLeague compared to [OP.GG](#)?** \*


36. What could be added to VisuaLeague so that it could facilitate the performed tasks? (If applicable) \*

37. Would you use VisuaLeague? \*  
*Mark only one oval.*

☐ Yes

☐ No

38. Justify the previous choice. \*

39. Extras \*

# References

- Andrienko, N., Andrienko, G., and Gatala, P. (2003). Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages & Computing*, 14(6):503–541.
- Bagrow, L. (2017). *History of cartography*. Routledge.
- Bojko, A. A. (2009). Informative or misleading? heatmaps deconstructed. In *International Conference on Human-Computer Interaction*, pages 30–39. Springer.
- Bowman, B., Elmqvist, N., and Jankun-Kelly, T. (2012). Toward visualization for games: Theory, design space, and patterns. *IEEE transactions on visualization and computer graphics*, 18(11):1956–1968.
- Carreiro, N. N. (2016). Técnicas de visualização para melhor o desempenho em jogos online. Master’s thesis, Faculdade de Ciências da Universidade de Lisboa.
- Chen, C. (2005). Top 10 unsolved information visualization problems. *IEEE computer graphics and applications*, 25(4):12–16.
- Chittaro, L. (2006). Visualizing information on mobile devices. *Computer*, 39(3):40–45.
- Coppock, J. T. and Rhind, D. W. (1991). The history of gis. *Geographical information systems: Principles and applications*, 1(1):21–43.
- Drachen, A. and Schubert, M. (2013). Spatial game analytics. In *Game Analytics*, pages 365–402. Springer.
- Feitosa, V. R., Maia, J. G., Moreira, L. O., and Gomes, G. A. (2015). Gamevis: Game data visualization for the web. In *Computer Games and Digital Entertainment (SBGames), 2015 14th Brazilian Symposium on*, pages 70–79. IEEE.
- Fekete, J.-D., Van Wijk, J. J., Stasko, J. T., and North, C. (2008). The value of information visualization. In *Information visualization*, pages 1–18. Springer.
- Gagné, A. R., El-Nasr, M. S., and Shaw, C. D. (2011). A deeper look at the use of telemetry for analysis of player behavior in rts games. In *International Conference on Entertainment Computing*, pages 247–257. Springer.

- Gatalsky, P., Andrienko, N., and Andrienko, G. (2004). Interactive analysis of event data using space-time cube. In *Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on*, pages 145–152. IEEE.
- Goldsberry, K. (2012). Courtvision: New visual and spatial analytics for the nba. In *2012 MIT Sloan Sports Analytics Conference*.
- Gonçalves, T., Vieira, P., Afonso, A. P., Carmo, M. B., and Moucho, T. (2018). Analysing player performance with animated maps. In *Proceedings of the 22nd International Conference Information Visualisation, IV 2018*, pages 103–109.
- Griffin, A. L., MacEachren, A. M., Hardisty, F., Steiner, E., and Li, B. (2006). A comparison of animated maps with static small-multiple maps for visually identifying space-time clusters. *Annals of the Association of American Geographers*, 96(4):740–753.
- Gudmundsson, J. and Horton, M. (2017). Spatio-temporal analysis of team sports. *ACM Computing Surveys (CSUR)*, 50(2):22.
- Gudmundsson, J. and Wolle, T. (2014). Football analysis using spatio-temporal tools. *Computers, Environment and Urban Systems*, 47:16–27.
- Hägerstraand, T. (1970). What about people in regional science? *Papers in regional science*, 24(1):7–24.
- Hamari, J. and Sjöblom, M. (2017). What is esports and why do people watch it? *Internet research*, 27(2):211–232.
- Harrower, M. (2003). Tips for designing effective animated maps. *Cartographic Perspectives*, 44:63–65.
- Harrower, M. and Fabrikant, S. (2008). The role of map animation for geographic visualization. *Geographic visualization: concepts, tools and applications*, pages 49–65.
- Jutzi, C. E. and Connelly, J. (2007). Gaming utilizing actual telemetry data. US Patent 7,211,000.
- Keim, D. A. (2002). Information visualization and visual data mining. *IEEE Transactions on Visualization & Computer Graphics*, 7(1):1–8.
- Khan, M. and Khan, S. S. (2011). Data and information visualization methods, and interactive mechanisms: A survey. *International Journal of Computer Applications*, 34(1):1–14.
- Kraak, M.-J. (2003). The space-time cube revisited from a geovisualization perspective. In *Proc. 21st International Cartographic Conference*, pages 1988–1996. Citeseer.

- Li, Q., Xu, P., Chan, Y. Y., Wang, Y., Wang, Z., Qu, H., and Ma, X. (2017). A visual analytics approach for understanding reasons behind snowballing and comeback in moba games. *IEEE transactions on visualization and computer graphics*, 23(1):211–220.
- Liu, S., Cui, W., Wu, Y., and Liu, M. (2014). A survey on information visualization: recent advances and challenges. *The Visual Computer*, 30(12):1373–1393.
- Loh, C. S., Sheng, Y., and Ifenthaler, D. (2015). *Serious Games Analytics: Methodologies for Performance Measurement, Assessment, and Improvement*. Springer, 1 edition.
- Lv, Z., Tek, A., Da Silva, F., Empereur-Mot, C., Chavent, M., and Baaden, M. (2013). Game on, science-how video game technology may help biologists tackle visualization challenges. *PloS one*, 8(3):e57990.
- Moura, D., el Nasr, M. S., and Shaw, C. D. (2011). Visualizing and understanding players' behavior in video games: discovering patterns and supporting aggregation and comparison. In *Proceedings of the 2011 ACM SIGGRAPH symposium on video games*, pages 11–15. ACM.
- Opach, T., Gołębiowska, I., and Fabrikant, S. I. (2014). How do people view multi-component animated maps? *The Cartographic Journal*, 51(4):330–342.
- Perin, C., Vuillemot, R., and Fekete, J.-D. (2013). Soccerstories: A kick-off for visual soccer analysis. *IEEE transactions on visualization and computer graphics*, 19(12):2506–2515.
- Pileggi, H., Stolper, C. D., Boyle, J. M., and Stasko, J. T. (2012). Snapshot: Visualization to propel ice hockey analytics. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2819–2828.
- Plaisant, C. (2004). The challenge of information visualization evaluation. In *Proceedings of the working conference on Advanced visual interfaces*, pages 109–116. ACM.
- Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE.
- Shrestha, A. (2014). *Visualizing Spatio-Temporal data*. PhD thesis, Georgia State University.
- Sifa, R., Drachen, A., and Bauckhage, C. (2018). Profiling in games: Understanding behavior from telemetry. *Social Interactions in Virtual Worlds: An Interdisciplinary Perspective*.
- Spence, R. (2001). *Information visualization*, volume 1. Springer.

- Stein, M., Janetzko, H., Lamprecht, A., Breitzkreutz, T., Zimmermann, P., Goldlücke, B., Schreck, T., Andrienko, G., Grossniklaus, M., and Keim, D. A. (2018). Bring it to the pitch: Combining video and movement data to enhance team sport analysis. *IEEE transactions on visualization and computer graphics*, 24(1):13–22.
- Thompson, C. (2007). Halo 3: How microsoft labs invented a new science of play. *Wired Magazine*, 15(9):15–09.
- Thompson, J. J., Blair, M. R., Chen, L., and Henrey, A. J. (2013). Video game telemetry as a critical tool in the study of complex skill learning. *PloS one*, 8(9):e75129.
- Tominski, C., Schulze-Wollgast, P., and Schumann, H. (2005). 3d information visualization for time dependent data on maps. In *Information Visualisation, 2005. Proceedings. Ninth International Conference on*, pages 175–181. IEEE.
- Vieira, P. (2017). Visualization of spatio-temporal information for personal performance analysis in games. Master’s thesis, Faculty of Science, University of Lisbon.
- Wagner, M. G. (2006). On the scientific relevance of esports. In *Proceedings of the 2006 International Conference on Internet Computing & Conference on Computer Games Development, ICOMP 2006, Las Vegas, Nevada, USA, June 26-29, 2006*, pages 437–442.
- Wallner, G. and Kriglstein, S. (2013). Visualization-based analysis of gameplay data—a review of literature. *Entertainment Computing*, 4(3):143–155.
- Wallner, G. and Kriglstein, S. (2015). Comparative visualization of player behavior for serious game analytics. In *Serious Games Analytics*, pages 159–179. Springer.
- Wright, T., Boria, E., and Breidenbach, P. (2002). Creative player actions in fps online video games: Playing counter-strike. *Game studies*, 2(2):103–123.
- Yang, P., Harrison, B. E., and Roberts, D. L. (2014). Identifying patterns in combat that are predictive of success in moba games. In *FDG*.
- Zammitto, V. (2008). Visualization techniques in video games. In *EVA*.